一般化された対数に基づく高速な独立成分分析法に関する研究

(13680465)

平成 13 年度～平成 14 年度科学研究費補助金（基盤研究 C）研究成果報告書

平成 15 年 3 月

松山 泰男

(早稲田大学理工学部教授)

はしがき

この報告書は，平成13年度～平成14年度まで，科学研究費補助金・基盤研究(C)(2)として，「一般化された対数に基づく高速な独立成分分析法に関する研究」を行った成果をまとめたものである．得られた成果は，下記のように項目化できる．

(1) 確率過程間の一般化距離である凸ダイバージェンスは，フィッシャー情報量やフィッシャースコアの拡張を与える．

(2) 凸ダイバージェンスから，一般化された対数を導出できる．

(3) 独立成分分析（ICA, Independent Component Analysis）のアルゴリズムを，凸ダイバージェンスの最小化から導くことができる．そしてこのアルゴリズムを f-ICA と命名した．

(4) f-ICA をソフトウェアとして実現する場合，過去の更新量を用いるモーメンタム法と，将来の予測値を利用するターボ法の２種類が存在する．

(5) どちらの方法も従来の相互情報量最小化法に比べて，数倍高速となる．特にモーメンタム法においては，必要なメモリー量の増加はわずかである．

(6) 通常の独立成分分析においては不可避であった順列不決定性の回避のために，先見知識を正則化項として注入する方法を与えた．

(7) 得られたソフトウェアシステムは高速かつコンパクトであるので，大規模データに対してもパーソナルコンピュータで処理できることとなった．

(8) 得られたアルゴリズムを人間の脳の fMRI 画像（磁気共鳴機能画像）に適用し，脳の機能マップを得た．

本研究は，以上のような成果を得て終了した．

研究組織

研究代表者：　松山 泰男（早稲田大学理工学部教授）

交付決定額（配分額）

（金額単位：千円）

	直接経費	間接経費	合計
平成13年度	2,200	0	2,200
平成14年度	1,300	0	1,300
総計	3,500	0	3,500

研究発表

(Ⅰ) 学会誌等

(1) Yasuo Matsuyama and Shuichiro Imahara, Independent component analysis by convex divergence: Applications to brain fMRI analysis, Proceedings of International Joint Conference on Neural Networks, Vol. 1, pp. 412-417, 2001.

(2) Yasuo Matsuyama, Naoto Katsumata and Shuichiro Imahara, Independent component analysis using convex divergence, Proceedings of International Conference on Neural Information Processing, Vol. 3, pp. 1173-1178, 2001

(3) Yasuo Matsuyama, Naoto Katsumata and Shuichiro Imahara, Convex divergence as a surrogate function for independence: The f-divergence ICA, Proceedings of 3rd International Conference on Independent Component Analysis and Signal Separation, Vol. 1, pp. 31-36, 2001.

(4) Y. Matsuyama, Shuichiro Imahara and Naoto Katsumata, Optimization transfer for computational learning: A hierarchy from f-ICA and alpha-EM to their offsprings, Proceedings of International Joint Conference on Neural Networks, Vol. 3, pp. 1883-1888, 2002.

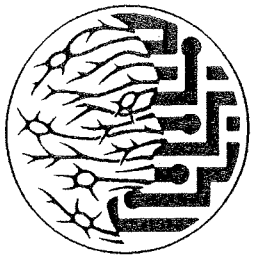
(5) Yasuo Matsuyama, Naoto Katsumata and Ryo Kawamura, Optimization transfer using convex divergence: f-ICA and alpha-EM algorithm with examples, Proceedings of International Symposium on Information Theory and Its Applications, Vol. 2, pp. 677-670, 2002.

(6) Y. Matsuyama and R. Kawamura, Supervised map ICA: Applications to brain functional MRI, International Conference on Neural Information Processing, Vol. 5, pp. 2259-2263, 2002.

(7) Y. Matsuyama, The α-EM algorithm: Surrogate likelihood maximization using α-logarithmic information measures, IEEE Transactions on Information Theory, Vol. 49, accepted, 2003.

(Ⅱ) 口頭発表

(1) Yasuo Matsuyama, Optimization transfer in learning algorithms: With examples on pattern extraction and emotional coding, Abstract of China-Japan-Korea Joint Workshop on Neurobiology and Neuroinformatics, invited talk, 2001.



IJCNN'01

Cosponsored by: The International Neural Network Society
The Neural Networks Council of IEEE

International Joint Conference
on
Neural Networks

Washington, DC
July 15-19, 2001

Independent Component Analysis by Convex Divergence Minimization: Applications to Brain fMRI Analysis

Yasuo Matsuyama and Shuichiro Imahara

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan.
{yasuo, shoe16}@wizard.elec.waseda.ac.jp

Abstract

A class of ICA algorithms (Independent Component Analysis) using a minimization of the convex divergence is presented. This is called the f-ICA. This algorithm is a super class of the minimum mutual information ICA and our own α-ICA. The following properties are obtained in this paper:
(i) The f-ICA can be implemented by both momentum and turbo methods. Their combination is also possible.
(ii) Formerly presented α-ICA can claim an equivalent form to the f-ICA if the design parameter α is chosen appropriately.
(iii) The f-ICA is much faster than the minimum mutual information ICA.
(iv) Additional complexity required to the divergence ICA is light. Therefore, this algorithm is applicable to large amount of data via conventional personal computers.
(v) Detection of human brain areas that strongly respond to moving objects is reported in this paper.

1 Introduction

ICA (Independent Component Analysis) is a method to find statistically independent components among measured data. There are many types of ICA algorithms. All methods are derived from optimization of surrogate functions for measuring the degree of independence. Examples of such optimization transfer are differential entropy, mutual information, kurtosis and cumulants [1], [2], [3], [4], [5], [6]. Convergence speeds are different. Memory requirements are also diverse. Because of versatility in surrogate functions, resulting degrees of independence are also different. There is an angle which shed light to the choice of a specific ICA. It is the flexibility to combine with other algorithms arising from computational intelligence studies. We expect that the min-mutual information approach [3], [4], [5] has a wide possibility in this direction. Therefore, we discuss extended versions of this type throughout the text.

It is found in our previous papers that the min-mutual information approach is a special case of the α-divergence minimization [7], [8], [9]. This is the α-ICA. The α-ICA has concrete merits on the speedup of convergence. Since the α-divergence is a special case of a general convex divergence, we start this paper with theoretical interest.

2 Convex Divergence and Independence

2.1 Convex Divergence

The convex divergence between two probability densities p and q are defined by the following equation [10].

$$\begin{aligned} D_f(p\|q) &= \int_{\mathcal{Y}} q(y) f(p(y)/q(y)) dy \\ &= \int_{\mathcal{Y}} p(y) g(q(y)/p(y)) dy \\ &= D_g(q\|p) \end{aligned} \tag{1}$$

Here, $\mathcal{Y}$ is a K-dimensional Euclidian space. The function $f(r)$, $r \in (0, \infty)$, is twice differentiable and convex. We choose $f(1) = 0$. Then, $g(r) = rf(1/r)$ is also twice differentiable and convex with $g(1) = 0$.

2.2 Mutual Information by Convex Divergence

Mutual information is a measure of independence among random variables. It measures a kind of distance between a joint probability density and a product probability density. Therefore, a generalized version of the mutual information using the convex divergence is defined as follows.

$$I_f(\wedge_{i=1}^K Y_i) \stackrel{\text{def}}{=} D_f\left(p(y_1, \cdots, y_K) \| \prod_{i=1}^K q_i(y_i)\right)$$

0-7803-7044-9/01/$10.00 ©2001 IEEE

ISCNN'2001, July, Washington DC

vol. 1

$$\stackrel{\text{def}}{=} D_f(p(y)\|q(y)) \tag{2}$$

Here, "$\bigwedge$" is used instead of the symbol ";" which appears in standard references [11]. Note that the α-divergence is the case of

$$f^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}). \tag{3}$$

In this case,

$$g^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(1 - r^{\frac{1+\alpha}{2}}). \tag{4}$$

Further special case of $\alpha = -1$ generates usual logarithmic mutual information by

$$f^{(-1)}(r) = r \log r \tag{5}$$

and

$$g^{(-1)}(r) = -\log r. \tag{6}$$

3 Derivation of the Convex Divergence ICA

3.1 Minimization of the Convex Divergence

Because of (1) and (2), the mutual information in terms of the convex divergence satisfies the following equality.

$$\begin{aligned} I_f(\textstyle\bigwedge_{i=1}^K Y_i) &= D_f(p(y)\|q(y)) \\ &= D_g(q(y)\|p(y)) = I_g(\textstyle\bigwedge_{i=1}^K Y_i) \end{aligned} \tag{7}$$

In the problem of ICA, we can observe a set of N random vectors.

$$x(t) = \text{col}[x_1(t), \cdots, x_K(t)], \quad (t = 1, \cdots, N). \tag{8}$$

We want to find a demixing matrix $W = \Lambda\Pi A^{-1}$ so that the components of

$$Wx(t) \stackrel{\text{def}}{=} y(t) = \text{col}[y_1(t), \cdots, y_K(t)] \tag{9}$$

are independent each other for every t. Note that A^{-1} is the original unknown demixing matrix such that

$$A^{-1}x(t) = s(t). \tag{10}$$

Here,

$$s(t) = \text{col}[s_1(t), \cdots, s_K(t)] \tag{11}$$

is unknown except that its components are independent. Λ is a nonsingular diagonal matrix and Π is a permutation matrix. Both matrices are also unknown.

The demixing matrix W can be estimated by minimizing the convex divergence (7). A gradient descent method can be obtained by differentiating this measure of independence. We found that it is easier to differentiate I_g than I_f because of the following form.

$$I_g(\textstyle\bigwedge_{i=1}^K Y_i) = \int_{\mathcal{X}} p(x) g\left(\frac{|W|q(y)}{p(x)}\right) dx \tag{12}$$

In this expression, the determinant $|W|$ appears at only one place. Then, the negative gradient is obtained as follows.

$$\begin{aligned} -\nabla I_g(\textstyle\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W} \\ &= \int_{\mathcal{X}} |W|q(y) g'\left(\frac{|W|q(y)}{p(x)}\right) \{W^{-T} - \varphi(y)x^T\}\, dx \\ &= -\nabla I_f(\textstyle\bigwedge_{i=1}^K Y_i) \end{aligned} \tag{13}$$

Here,

$$-\varphi(y) = \text{col}\left[\frac{q_1'(y_1)}{q_1(y_1)}, \ldots, \frac{q_K'(y_K)}{q_K(y_K)}\right]. \tag{14}$$

Then, a simple update equation is

$$W(t+1) = W(t) + \Delta_f W(t) \tag{15}$$

with

$$\Delta_f W(t) = \rho_t \{-\nabla I_f(\textstyle\bigwedge_{i=1}^K Y_i)\}_{W=W(t)} \tag{16}$$

Here, t is the index for iteration and ρ_t is a learning rate.

3.2 Simplification by Natural Gradient

Similarly to [12] and [7] $\sim$ [9], we can apply a natural gradient to the update term. This is to multiply CW^TW from the right. The constant C will be given below.

$$\begin{aligned} -\tilde{\nabla} I_g(\textstyle\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W}(CW^TW) \\ &= -C \int_{\mathcal{X}} q(y) g'\left(\frac{|W|q(y)}{p(x)}\right)(I - \varphi(y)x^TW^T)|W|dx\, W \\ &= -C \int_{\mathcal{Y}} q(y) g'\left(\frac{q(y)}{p(y)}\right)(I - \varphi(y)y^T)dy\, W. \end{aligned} \tag{17}$$

Note that

$$qg'(q/p) = -g''(1)p + \{g'(1) + g''(1)\}q + o(1). \tag{18}$$

We can choose a scaling factor C such that

$$C = -\frac{g''(1)}{g'(1)} = \frac{f''(1)}{f'(1)} = \frac{d}{dr}[\log f'(r)]_{r=1}. \tag{19}$$

Then, we have

$$\begin{aligned} -\frac{\partial I_g}{\partial W}\left(-\frac{g''(1)}{g'(1)}W^TW\right) &= g''(1)\left\{I + \frac{g''(1)}{g'(1)}E_{p(y)}[\varphi(y)y^T]\right. \\ &\quad \left. - \left(1 + \frac{g(1)''}{g'(1)}\right)E_{q(y)}[\varphi(y)y^T]\right\}W + o(1). \end{aligned} \tag{20}$$

This leads to the following equation.

$$
\begin{aligned}
&-\frac{\partial I_f}{\partial W}\left(\frac{f''(1)}{f'(1)}W^TW\right)\\
&=-\frac{\partial I_g}{\partial W}\left(-\frac{g''(1)}{g'(1)}W^TW\right)\\
&=f''(1)\left[\frac{f''(1)}{f'(1)}\left\{I-E_{p(y)}[\varphi(y)y^T]\right\}W\right.\\
&\quad\left.+\left(1-\frac{f''(1)}{f'(1)}\right)\left\{I-E_{q(y)}[\varphi(y)y^T]\right\}W\right]+o(1)
\end{aligned}
\tag{21}
$$

3.3 Comparison with Minimum Mutual Information ICA

Because of (3) and (5), we have

$$f'(1)=\frac{df^{(\alpha)}(r)}{dr}|_{r=1}=\frac{2}{1-\alpha}=-g'(1) \tag{22}$$

and

$$f''(1)=\frac{d^2f^{(\alpha)}(r)}{dr^2}|_{r=1}=1=g''(1). \tag{23}$$

Therefore,

$$\frac{f''(1)}{f'(1)}=\frac{1-\alpha}{2}=-\frac{g''(1)}{g'(1)} \tag{24}$$

and

$$1-\frac{f''(1)}{f'(1)}=\frac{1+\alpha}{2}=1+\frac{g''(1)}{g'(1)}. \tag{25}$$

Since $\alpha=-1$ is the case of the logarithmic mutual information, the term (25) in (20) and (21) varnishes. This is the case of [3], [4] and [5]. We comment here that the α-ICA essentially *spans* the convex divergence ICA because of the expression (20).

4 Realization of the Convex Divergence ICA

4.1 Update Equation

The update equation is

$$W(t+1)=W(t)+\tilde{\Delta}_fW(t). \tag{26}$$

Here,

$$\tilde{\Delta}_fW(t)=\rho_t\left\{-\tilde{\nabla}I_f(\wedge_{i=1}^K Y_i)\right\}_{W=W(t)}. \tag{27}$$

We call this method the f-ICA.

By observing (20) and (21), we have the following interpretation on the update terms.

(i) The coefficient $f''(1)$ can be absorbed in the learning rate ρ_t.

(ii) The coefficients

$$m_f\stackrel{\text{def}}{=}\frac{f''(1)}{f'(1)} \tag{28}$$

and

$$\bar{m}_f\stackrel{\text{def}}{=}1-\left\{\frac{f''(1)}{f'(1)}\right\}=1-m_f \tag{29}$$

play the same roles as $\frac{1-\alpha}{2}$ and $\frac{1+\alpha}{2}$ of the α-ICA algorithms.

Therefore, the interpretation of p and q in terms of the shift of iteration index remains valid.

4.2 Causal Realization as the Momentum f-ICA

First, we observe that $q(y)$ is the target function of $p(y)$ such that

$$q(y)=\lim_{t\to\infty}p^{(t)}(y) \tag{30}$$

under an appropriate convergence criterion such as the vague convergence. Here, t is the index for the iteration count. Then, there is a causal approximation at the t-th iteration such that

$$q(y)\approx p^{(t)}(y)\quad\text{and}\quad p(y)=p^{(t-\tau)}(y). \tag{31}$$

By this approximation, we have the following sample-based learning algorithm.

[Momentum f-ICA: Algorithm (a)]

If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$
\begin{aligned}
&\tilde{\Delta}_fW(t)=\tilde{\Delta}W(t)+\mu_f\tilde{\Delta}W(t-\tau)\\
&=\rho_t\left[\{I-\varphi(y(t))y(t)^T\}W(t)\right.\\
&\quad\left.+\mu_f\{I-\varphi(y(t-\tau))y(t-\tau)^T\}W(t-\tau)\right]
\end{aligned}
\tag{32}
$$

Here,

$$\mu_f=m_f/\bar{m}_f \tag{33}$$

Thus, we added a momentum term $\tilde{\Delta}W(t-\tau)$ weighted by μ_f. Note that the case of $\mu_f=\frac{1-\alpha}{1+\alpha}$ corresponds to the α-ICA [7] $\sim$ [9]. Further special case of $\alpha=1$, i.e., $\mu_f=0$ is

$$\tilde{\Delta}_fW(t)=\tilde{\Delta}W(t) \tag{34}$$

which is the plain minimum mutual information method of [5].

4.3 Non-Causal Realization as the Turbo f-ICA

There is a non-causal approximation at the t-th iteration such that

$$q(y)\approx p^{(t+\tau)}(y)\quad\text{and}\quad p(y)=p^{(t)}(y). \tag{35}$$

This is more natural than the momentum f-ICA since $p(y)$ has the present index. Then, we have the following sample-based learning algorithm.

[Turbo f-ICA (Look-ahead f-ICA): Algorithm (b)]

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta} W(t) + \nu_f \tilde{\Delta} W(t+\tau) \\ &= \rho_t \lceil \{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \nu_f \{I - \varphi(\hat{y}(t+\tau))\hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \rceil \end{aligned} \tag{36}$$

Here,

$$\nu_f = 1/\mu_f = \bar{m}_f / m_f \tag{37}$$

The look-ahead terms $\hat{W}(t+\tau)$ and $\hat{y}(t+\tau)$ are estimations of $W(t+\tau)$ and $y(t+\tau)$ using the usual log-version. Thus, we added a predicted term $\tilde{\Delta}\hat{W}(t+\tau)$ weighted by ν_f.

We comment here that there is a duality between Equations (32) and (36). The turbo f-ICA of $\nu_f = 0$ is the existing case (34). This is an inherited property from the f-divergence. We also note in advance that $\tau = 1$ works effectively enough for both causal and non-causal methods in spite of the asymptotic relationship (30).

4.4 Orthogonal f-ICA

Amari et al. [13] introduced an orthogonal ICA which is expected to suppress zero-power fake signals. The idea is to find an update term, say $\tilde{\Delta}^+ W$, which is orthogonal to $\tilde{\Delta} W$ so that

$$< \tilde{\Delta} W, \tilde{\Delta}^+ W >_W = 0. \tag{38}$$

Such an update term $\tilde{\Delta}^+ W$ is obtained as follows. Let

$$\Lambda = \mathrm{diag}\,[\lambda_i]_{i=1}^K \tag{39}$$

is a non-singular diagonal matrix. Let

$$W + \tilde{\Delta} W = (I + d\Lambda) W. \tag{40}$$

Then,

$$\tilde{\Delta}^+ W = \rho\{\Lambda - \varphi(y) y^T\} W \tag{41}$$

such that

$$\Lambda = \mathrm{diag}\,[\varphi_i(y_i) y_i]_{i=1}^K. \tag{42}$$

Therefore, we have the following four types of orthogonal f-ICA algorithms.

[Orthogonal momentum f-ICA: Algorithm (c)]
If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned}\tilde{\Delta}_f^+ W(t) &= \tilde{\Delta}^+ W(t) + \mu_f \tilde{\Delta}^+ W(t-\tau) \\ &= \rho_t \lceil \{\Lambda(t) - \varphi(y(t))y(t)^T\} W(t) + \mu_f \\ &\quad \times \{\Lambda(t-\tau) - \varphi(y(t-\tau))y(t-\tau)^T\} W(t-\tau) \rceil \end{aligned} \tag{43}$$

[Turbo (Look-ahead) methods: Algorithms (d0) $\sim$ (d2)]
The update term is as follows.

$$\begin{aligned}\tilde{\Delta}_f^+ W(t) &= \tilde{\Delta}^+ W(t) + \nu_f \tilde{\Delta}^+ \hat{W}(t+\tau) \\ &= \rho_t \lceil \{\Gamma(t) - \varphi(y(t))y(t)^T\} W(t) + \nu_f \\ &\quad \times \{\hat{\Omega}(t+\tau) - \varphi(\hat{y}(t+\tau))\hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \rceil \end{aligned} \tag{44}$$

Here, the matrices $\Gamma(t)$ and $\hat{\Omega}(t+\tau)$ are as follows.

(d0) $\Gamma(t) = \Lambda(t)$ and $\hat{\Omega}(t+\tau) = \hat{\Lambda}(t+\tau)$ give a purely orthogonal turbo f-ICA.
(d1) $\Gamma(t) = I$ and $\hat{\Omega}(t+\tau) = \hat{\Lambda}(t+\tau)$ give a hybrid turbo f-ICA of type I.
(d2) $\Gamma(t) = \Lambda(t)$ and $\hat{\Omega}(t+\tau) = I$ give a hybrid turbo f-ICA of type II.

4.5 Combination of Momentum and Turbo f-ICA's

The momentum f-ICA exploits one-step past; (32) and (43) with $\tau = 1$. The turbo f-ICA uses one-step future by prediction; (36) and (44) with $\tau = 1$). Then, one expects that both methods are incorporated to a single ICA algorithm. Thus,

$$\tilde{\Delta}_f W(t) = \tilde{\Delta} W(t) + \mu_t \tilde{\Delta} \hat{W}(t-\tau) + \nu_t \tilde{\Delta} \hat{W}(t+\tau) \tag{45}$$

can be used as the update equation.

4.6 Speed Evaluation

4.6.1 Experimental Evaluation

For sample-based learning, the expectation $E[\cdot]$ is replaced by $\frac{1}{T}\sum_{i=1}^T [\cdot]$ where T is the number of samples in a selected window. This is a semi-batch mode. The case of $T = 1$ is the incremental learning, If T is equal to the number of the whole data elements, the method becomes a full batch learning.

We chose mixtures of time series [5] as benchmarking problems. The non-linearity of $\varphi(y) = y^3$ [1] was used. The convergence speed was measured by the cross-talking error [5] which checks the closeness of the matrix WA to $\Lambda\Pi$. Table I summarizes the convergence speed measured by iterations.

Table I
Comparison of iteration counts for ICA's.

plain MMI	momentum	turbo	m+t
280	120	50	42

Here, "m+t" stands for the combination of "momentum" and "turbo." It is important to note the following.

(a) Consider a case of plain minimum-mutual information ICA. If we were a priori given the largest $\{\rho_t\}_{t\geq 1}$ which gives convergence of $\{W(t)\}_{t\geq 1}$, this case would give an almost optimally fast speed. But, such a series of constants $\{\rho_t\}_{t\geq 1}$ can never be given before learning iterations. Therefore, we have to fix $\rho_t = \rho$. Even so, a limiting large ρ can not be found in advance because this figure also depends on unknown probability densities. Therefore, we need speedup methods which are compatible with safe choices of ρ. The class of f-ICA algorithms discussed in this paper is a viable candidate. The numbers $\nu_f = 5.0$ and $\mu_f = 2.0$ are recommended choices.

(b) Even in the neighborhood of the convergence region, use of the momentum term is still recommended. This is because the momentum method requires only very few computational complexity.

4.6.2 Evaluation from the Convex Divergence

Because of (13) and (17), the speed of the f-ICA against the log-ICA (minimum mutual information ICA) can be evaluated by (18). Further manipulation gives the following.

$$q(y)g'(\tfrac{q(y)}{p(y)}) = -g''(1)p(y)\left[1 + \frac{1+\frac{g''(1)}{g'(1)}\frac{q(y)}{p(y)}}{-\frac{g''(1)}{g'(1)}}\right] + o(1)$$

$$= -g''(1)p(y)\left[1 + \frac{1-C}{C}\frac{q(y)}{p(y)}\right] + o(1) \quad (46)$$

Therefore, $0 < C < 1$ gives faster convergence by the rate of $1 + (1 - C)/C$.

5 Applications to Brain fMRI Maps

5.1 Use of Source Data

The test data of the brain fMRI is a set of $S \times T$ two-dimensional images. Here, $S = 7$ is the number of slices of a human head. Each slice contains 128×128 pixels as a two-dimensional array. $T = 68$ is the number of samples in the time axis.

Three visual patterns were shown to a tested person. They are (r) a dark background with a small red cross at the center, (s) a still image with many white squares located randomly, and (m) two groups of moving squares in opposite directions each other. Our goal is to find (i) a set of independent maps, and (ii) its associated activation patterns which are relevant to recognizing the moving image.

For the experiment, we picked up a data set of "*ssmmssmm*." Therefore, $x(t)$, $(t = 1, \ldots, 128 \times 128)$, is a set of eight-dimensional column vectors corresponding to "*ssmmssmm*." The index t stands for one of pixel positions in 128×128 region. Thus, $y(t)$, $(t = 1, \ldots, 128 \times 128)$, give eight images of extracted maps. Since

$$x(t) = W^{-1}y(t) \stackrel{\text{def}}{=} Uy(t), \quad (47)$$

the column vectors in the matrix U can be interpreted as activation patterns of the eight maps.

Fig. 1 (left) is an activation pattern which shows "low-low-high-high-low-low-high-high." Fig. 1 (right) is its corresponding map which is a superposition of active areas (black) and anatomical data (half tone). This experiment was successful by using $\varphi(y) = tanh(cy)$. The black horizontal line at the one third rear of the section is a zero-frequency artifact. Thus, we found a very active region at the rear part of the right hemisphere for this tested person (a young male adult).

6 Concluding Remarks

In this paper, we presented the f-ICA which is derived from the minimization of the convex divergence. The derivation started from a theoretical interest. But, resulting software implementations show remarkable speed as a gradient descent type. This was because effective use of past and/or future data. Additional software complexity is light.

In the application part, we addressed a problem of finding task-related fMRI brain maps. An activation pattern corresponding to the recognition of the moving image was successfully found.

Acknowledgments

The authors are quite thankful to Dr. R. Allen Waggoner, Dr. Keiji Tanaka and Dr. Hiroshige Takeichi of RIKEN BRI for permitting them to try out the test data set.

The authors are also indebted to the Grant-in-Aid for Scientific Research as well as the High Technology and New Technology Projects of Waseda University.

References

[1] C. Jutten and J. Herault, Blind separation of sources, Part I: An adaptive algorithm based

on neuromimetic architecture, Signal Processing, vol. 24, pp. 1-20, 1991.

[2] J.-F. Cardoso and A. Souloumiac, Blind beamforming for non Gaussian signals, IEE Proceedings-F, vol. 140, No. 6, pp. 362-370, 1993.

[3] P. Comon, Independent component analysis, A new concept?, Signal Processing, vol. 36, pp. 287-314.

[4] A.J. Bell and T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[5] H.H. Yang and S. Amari, Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

[6] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks, vol. 10, pp. 626-634, 1999.

[7] Y. Matuyama, N. Katsumata, Y. Suzuki and S. Imahara, The α-ICA algorithm, Proc. ICA2000, pp. 297-302, 2000.

[8] Y. Matsuyama, T. Niimoto, N. Katsumata, Y. Suzuki and S. Furukawa, α-EM algorithm and α-ICA learning based upon extended logarithmic information measures, Proc. IJCNN2000, vol. III, pp. 351-356, 2000.

[9] Y. Matsuyama and S. Imahara, The α-ICA algorithm and brain map distillation from fMRI images, Proc. ICONIP2000, vol. 2, pp. 708-713, 2000.

[10] I. Csiszár, Information-type measures of difference of probability distributions and indirect observations, Studia Svi. Math. Hungarica, vol. 2, pp. 299-318, 1967.

[11] T.M. Cover and J.A. Thomas, *Elements of Information Theory,* Wiley, 1991.

[12] S. Amari, Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

[13] S. Amari T-P. Chen and A.J. Cichocki, Non-holonomic constraints in learning blind source separation, Proc. ICONIP'97, vol. 1, pp. 633-636, 1997.

[14] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski, Spacially independent activity patterns in functional MRI data during the Stroop color-naming task, Proc. Natl. Acad. Sci. USA, vol. 95, pp. 803-810, 1998.

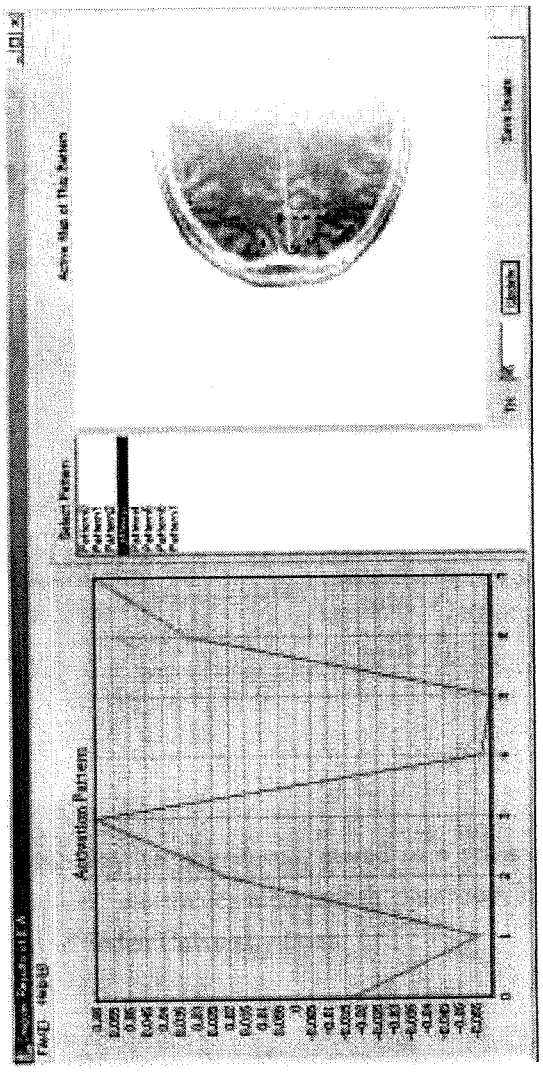


Figure 1: Activation pattern for "*ssmmssmm*" and its associated map.

8th International Conference on Neural Information Processing

ICONIP2001

Fudan University Press
579 Guoquan Rd. Shanghai, 200433, China

(2001 Shanghai International Industrial FAIR Science Technology Forum)

8th International Conference on Neural Information Processing

ICONIP-2001

Proceedings

Volume 3

Sponsored by

Asia-Pacific Neural Network Assembly (APNNA)

China Neural Network Council (CNNC)

Fudan University

Shanghai Association for Science & Technology (SAST)

Co-sponsored by

IEEE Beijing Section, IEEE NN Council, INNS

Organized by

Center for Brain Science Research, Fudan University

Shanghai Society of Biophysics

Supported by

The Ministration of Science and Technology of China

The National Natural Science Foundation of China

Mr. F. S. Lin, Director of Fudan University, Chairman of Milo's Knitwear (Hong Kong)

The Third World Academy of Science (TWAS)

November

Independent Component Analysis Using Convex Divergence

Yasuo Matsuyama, Naoto Katsumata and Shuichiro Imahara

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan.
{yasuo, katsu, shoe16}@wizard.elec.waseda.ac.jp

Abstract

The convex divergence is used as a surrogate function for obtaining a class of ICA algorithms (Independent Component Analysis) called the f-ICA. The convex divergence is a super class of α-divergence, which is a further upper family of Kullback-Leibler divergence or mutual information. Therefore, the f-ICA contains the α-ICA and the minimum mutual information ICA. In addition to theoretical interest of generalization, the f-ICA contains a subset faster than the minimum mutual information ICA. It is found that this speed control is equivalent to the α-ICA. Finally, applications to brain fMRI map's distillation is presented.

1 Introduction

The independent component analysis (ICA) is a method to separate statistically independent components among mixture of source data. Degree of independence is judged through various surrogate functions including kurtosis, cumulants, differential entropy (simply, entropy hereafter), and average mutual information (simply, mutual information hereafter) [1], [2], [3], [4], [5], [6]. Convergence speeds are different. Memory requirements are also diverse. Because of versatility in surrogate functions, resulting degrees of independence are also different. In this paper, we focus on generalization of the minimum mutual information ICA. This is because we expect that the min-mutual information approach [3], [4], [5] will have possibility to easily combine with other computational intelligence methods.

Organization of this paper is as follows. First, we start with theoretical discussions. Next, methods to implement as computer software are discussed. Then, relationship to our α-ICA [7], [8], [9] is discussed. Finally, speed evaluation and applications to brain fMRI map distillation are tried.

2 Convex Divergence, Subclasses and Independence

2.1 Convex Divergence and Its Subclasses

The convex divergence between two probability densities p and q is defined by the following equation [10].

$$D_f(p\|q) = \int_{\mathcal{Y}} q(y) f(p(y)/q(y)) dy \quad (1)$$

$$= \int_{\mathcal{Y}} p(y) g(q(y)/p(y)) dy \quad (2)$$

$$= D_g(q\|p) \geq g(1) = f(1). \quad (3)$$

Here, $\mathcal{Y}$ is a K-dimensional Euclidian space. The function $f(r)$, $r \in (0, \infty)$, is convex and twice differentiable. $g(r)$ satisfies $g(r) = rf(1/r)$ so that it is also convex and twice differentiable. The inequality part of (3) is the equality if and only if $p(y) = q(y)$, y-a.e.

If analytical solutions are required [8], some structures are necessary on the functions $f(r)$ and $g(r)$. A useful class of functions satisfies the following equality.

$$f(xy) = kf(x)f(y) \quad (4)$$

Such a class is expressed by

$$f(r) = c(\beta)\, r^{\beta}. \quad (5)$$

Here, β and $c(\beta)$ should have a relationship so that $f(r)$ be a convex function. If we choose $f(1) = g(1) = 0$, then

$$f^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}), \quad (6)$$

and

$$g^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(1 - r^{\frac{1+\alpha}{2}}) \quad (7)$$

are such convex functions for $\alpha \in (-\infty, \infty)$. It is important to note that there are further special cases with $\alpha = -1$ which relate to the logarithm:

$$f^{(-1)}(r) = r \log r \quad (8)$$

and

$$g^{(-1)}(r) = -\log r. \tag{9}$$

2.2 Mutual Information by Convex Divergence

The convex divergence gives a definition of mutual information. The mutual information gives a degree of independence among random variables by measuring a distance between a joint probability density and a product probability density. Thus, a generalized version of the mutual information using the convex divergence is defined as follows.

$$\begin{aligned} I_f(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} D_f\left(p(y_1,\cdots,y_K)\|\prod_{i=1}^K q_i(y_i)\right) \\ &\stackrel{\text{def}}{=} D_f(p(y)\|q(y)) \end{aligned} \tag{10}$$

Here, "$\bigwedge$" is used instead of the symbol ";" which appears in standard references [11].

3 Derivation of the f-ICA

3.1 Minimization of the Convex Divergence

Because of (1) $\sim$ (3), the mutual information in terms of the convex divergence satisfies the following equality.

$$\begin{aligned} I_f(\bigwedge_{i=1}^K Y_i) &= D_f(p(y)\|q(y)) \\ &= D_g(q(y)\|p(y)) = I_g(\bigwedge_{i=1}^K Y_i) \end{aligned} \tag{11}$$

In the problem of ICA, we can observe a set of N random vectors.

$$x(t) = \mathrm{col}[x_1(t),\cdots,x_K(t)], \quad (t=1,\cdots,N). \tag{12}$$

We want to find a demixing matrix $W = \Lambda\Pi A^{-1}$ so that the components of

$$Wx(t) \stackrel{\text{def}}{=} y(t) = \mathrm{col}[y_1(t),\cdots,y_K(t)] \tag{13}$$

are independent each other for every t. Note that A^{-1} is the original unknown demixing matrix such that

$$A^{-1}x(t) = s(t). \tag{14}$$

Here,

$$s(t) = \mathrm{col}[s_1(t),\cdots,s_K(t)] \tag{15}$$

is unknown except that its components are independent. Λ is a nonsingular diagonal matrix and Π is a permutation matrix. Both matrices are also unknown.

The demixing matrix W can be estimated by minimizing the convex divergence (11). A gradient descent method can be obtained by differentiating this measure of independence. We found that it is easier to differentiate I_g than I_f because of the following form:

$$I_g(\bigwedge_{i=1}^K Y_i) = \int_{\mathcal{X}} p(x) g\left(\frac{|W|q(y)}{p(x)}\right) dx \tag{16}$$

In this expression, the determinant $|W|$ appears at only one place. Here, we used (11) and

$$dy = |W|dx \tag{17}$$

as well as

$$p(y)dy = p(x)dx. \tag{18}$$

Then, the negative gradient is obtained as follows.

$$\begin{aligned} -\nabla I_g(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W} \\ &= \int_{\mathcal{X}} |W|q(y) g'\left(\tfrac{|W|q(y)}{p(x)}\right)\left\{W^{-T} - \varphi(y)x^T\right\} dx \\ &= -\nabla I_f(\bigwedge_{i=1}^K Y_i) \end{aligned} \tag{19}$$

Here,

$$g'(r) = \tfrac{d}{dr} g(r). \tag{20}$$

and

$$-\varphi(y) = \mathrm{col}\left[\tfrac{q_1'(y_1)}{q_1(y_1)},\ldots,\tfrac{q_K'(y_K)}{q_K(y_K)}\right], \tag{21}$$

where

$$q_i'(y_i) = \tfrac{d}{dy} q(y)|_{y=y_i}. \tag{22}$$

Then, a simple update equation is

$$W(t+1) = W(t) + \Delta_f W(t) \tag{23}$$

with

$$\begin{aligned} \Delta_g W(t) &= \rho_t\left\{-\nabla I_g(\bigwedge_{i=1}^K Y_i)\right\}_{W=W(t)} \\ &= \rho_t\left\{-\nabla I_f(\bigwedge_{i=1}^K Y_i)\right\}_{W=W(t)} \\ &= \Delta_f W(t). \end{aligned} \tag{24}$$

Here, t is the index for iteration and ρ_t is a learning rate.

3.2 Simplification by Natural Gradient

Similarly to [12] and [7] $\sim$ [9], we can apply a natural gradient to the update term. This is to multiply CW^TW from the right. The constant C will be given later.

$$\begin{aligned} -\tilde{\nabla} I_g(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W}(CW^TW) \\ &= -C\int_{\mathcal{X}} q(y) g'\left(\tfrac{|W|q(y)}{p(x)}\right)(I - \varphi(y)x^TW^T)|W|dx\, W \\ &= -C\int_{\mathcal{Y}} q(y) g'\left(\tfrac{q(y)}{p(y)}\right)(I - \varphi(y)y^T)dy\, W. \end{aligned} \tag{25}$$

An important next step is how to evaluate the core of the integrand of (25). It is important to observe

$$qg'(q/p) = -g''(1)p + \{g'(1) + g''(1)\}q + o(1) \quad (26)$$

around $p \approx q$. We can choose a scaling factor C such that

$$C = -\frac{g''(1)}{g'(1)} = \frac{f''(1)}{f'(1)} = \frac{d}{dr}[\log f'(r)]_{r=1}. \quad (27)$$

Then, we have

$$\begin{aligned} &-\frac{\partial I_g}{\partial W}\left(-\frac{g''(1)}{g'(1)}W^TW\right) \\ &\quad = g''(1)\left\{I + \frac{g''(1)}{g'(1)}E_{p(y)}[\varphi(y)y^T]\right. \\ &\quad\quad \left. - \left(1 + \frac{g(1)''}{g'(1)}\right)E_{q(y)}[\varphi(y)y^T]\right\}W + o(1). \quad (28) \end{aligned}$$

This leads to the following equation.

$$\begin{aligned} &-\frac{\partial I_f}{\partial W}\left(\frac{f''(1)}{f'(1)}W^TW\right) \\ &= -\frac{\partial I_g}{\partial W}\left(-\frac{g''(1)}{g'(1)}W^TW\right) \\ &= f''(1)\left[\frac{f''(1)}{f'(1)}\left\{I - E_{p(y)}[\varphi(y)y^T]\right\}W\right. \\ &\quad \left. + \left(1 - \frac{f''(1)}{f'(1)}\right)\left\{I - E_{q(y)}[\varphi(y)y^T]\right\}W\right] + o(1) \quad (29) \end{aligned}$$

3.3 Comparison with Minimum Mutual Information ICA Using α-Version

Because of (6) and (7), we have

$$f'(1) = \frac{df^{(\alpha)}(r)}{dr}|_{r=1} = \frac{2}{1-\alpha} = -g'(1) \quad (30)$$

and

$$f''(1) = \frac{d^2f^{(\alpha)}(r)}{dr^2}|_{r=1} = 1 = g''(1). \quad (31)$$

Therefore,

$$\frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} = \frac{1-\alpha}{2} \quad (32)$$

and

$$1 - \frac{f''(1)}{f'(1)} = 1 + \frac{g''(1)}{g'(1)} = \frac{1+\alpha}{2} \quad (33)$$

Since $\alpha = -1$ is the case of the logarithmic mutual information, the term (33) in (28) and (29) varnishes. This is the case of [3], [4] and [5].

The α-ICA uses the design parameter α such that

$$-1 \leq \alpha < 1. \quad (34)$$

We comment here that the α-ICA essentially *spans* the f-ICA because of the expressions (28) and (29).

3.4 Evaluation from the Convex Divergence

Because of (19) and (25), the speed of the f-ICA compared with the log-ICA (minimum mutual information ICA) can be evaluated using (26). Further manipulation gives the following.

$$\begin{aligned} q(y)g'\left(\frac{q(y)}{p(y)}\right) &= -g''(1)p(y)\left[1 + \frac{1+\frac{g''(1)}{g'(1)}\frac{q(y)}{p(y)}}{-\frac{g''(1)}{g'(1)}}\right] + o(1) \\ &= -g''(1)p(y)\left[1 + \frac{1-C}{C}\frac{q(y)}{p(y)}\right] + o(1) \quad (35) \end{aligned}$$

Therefore,

$$0 < C \leq 1 \quad (36)$$

is the range corresponding to (34). The region of inequality in (36) gives faster convergence by the rate of $1 + \frac{1-C}{C}\frac{q}{p}$. Note that $C = 1$ is the case of the minimum mutual information ICA.

4 Realization of the Convex Divergence ICA

4.1 Update Equation

The update equation is

$$W(t+1) = W(t) + \tilde{\Delta}_f W(t). \quad (37)$$

Here,

$$\tilde{\Delta}_f W(t) = \rho_t \left\{-\tilde{\nabla} I_f(\Lambda_{i=1}^K Y_i)\right\}_{W=W(t)} \quad (38)$$

We call this method the f-ICA.

By observing (29), we have the following interpretation on the update terms.

(i) The coefficient $f''(1)$ can be absorbed in the learning rate ρ_t.

(ii) The coefficients

$$m_f \stackrel{\text{def}}{=} \frac{f''(1)}{f'(1)} \quad (39)$$

and

$$\bar{m}_f \stackrel{\text{def}}{=} 1 - \left\{\frac{f''(1)}{f'(1)}\right\} = 1 - m_f \quad (40)$$

play the same roles as $\frac{1-\alpha}{2}$ and $\frac{1+\alpha}{2}$ of the α-ICA algorithms.

Therefore, the interpretation of p and q in terms of the shift of iteration index remains valid.

4.2 Causal Realization as the Momentum f-ICA

First, we observe that $q(y)$ is the target function of $p(y)$ such that

$$q(y) = \lim_{t\to\infty} p^{(t)}(y) \tag{41}$$

under an appropriate convergence criterion such as the vague convergence. Here, t is the index for the iteration count. Then, there is a causal approximation at the t-th iteration such that

$$q(y) \approx p^{(t)}(y) \quad \text{and} \quad p(y) \approx p^{(t-\tau)}(y). \tag{42}$$

By this approximation, we have the following sample-based learning algorithm.

[Momentum f-ICA: Algorithm (a)]

If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta}W(t) + \mu_f \tilde{\Delta}W(t-\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \mu_f \{I - \varphi(y(t-\tau))y(t-\tau)^T\} W(t-\tau) \Big]\end{aligned} \tag{43}$$

Here,

$$\mu_f = m_f / \bar{m}_f \tag{44}$$

Thus, we added a momentum term $\tilde{\Delta}W(t-\tau)$ weighted by μ_f. Note that the case of $\mu_f = \frac{1-\alpha}{1+\alpha}$ corresponds to the α-ICA [7] $\sim$ [9]. Further special case of $\alpha = 1$, i.e., $\mu_f = 0$ is

$$\tilde{\Delta}_f W(t) = \tilde{\Delta}W(t) \tag{45}$$

which is the plain minimum mutual information method of [5].

4.3 Non-Causal Realization as the Turbo f-ICA

There is a non-causal approximation at the t-th iteration such that

$$q(y) \approx p^{(t+\tau)}(y) \quad \text{and} \quad p(y) \approx p^{(t)}(y). \tag{46}$$

This is more natural than the momentum f-ICA since $p(y)$ has the present index. Then, we have the following sample-based learning algorithm.

[Turbo (Look-ahead) f-ICA): Algorithm (b)]

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta}W(t) + \nu_f \tilde{\Delta}W(t+\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \nu_f \{I - \varphi(\hat{y}(t+\tau))\hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \Big]\end{aligned} \tag{47}$$

Here,

$$\nu_f = 1/\mu_f = \bar{m}_f / m_f \tag{48}$$

The look-ahead terms $\hat{W}(t+\tau)$ and $\hat{y}(t+\tau)$ are estimations of $W(t+\tau)$ and $y(t+\tau)$ using the usual log-version. Thus, we added a predicted term $\tilde{\Delta}\hat{W}(t+\tau)$ weighted by ν_f.

We comment here that there is a duality between Equations (43) and (47). The turbo f-ICA of $\nu_f = 0$ is the existing case (45). This is an inherited property from the f-divergence. We also note in advance that $\tau = 1$ works effectively enough for both causal and non-causal methods in spite of the asymptotic relationship (41).

4.4 Orthogonal f-ICA

Amari et al. [13] introduced an orthogonal ICA which is expected to suppress zero-power fake signals. The idea is to find an update term, say $\tilde{\Delta}^+ W$, which is orthogonal to $\tilde{\Delta}W$ so that

$$< \tilde{\Delta}W, \tilde{\Delta}^+ W >_W = 0. \tag{49}$$

Such an update term $\tilde{\Delta}^+ W$ is obtained as follows. Let

$$\Lambda = \mathrm{diag}\,[\lambda_i]_{i=1}^K \tag{50}$$

is a non-singular diagonal matrix. Let

$$W + \tilde{\Delta}W = (I + d\Lambda)W. \tag{51}$$

Then,

$$\tilde{\Delta}^+ W = \rho\{\Lambda - \varphi(y)y^T\}W \tag{52}$$

such that

$$\Lambda = \mathrm{diag}\,[\varphi_i(y_i)y_i]_{i=1}^K. \tag{53}$$

We can obtain four types of orthogonal f-ICA algorithms as is given in [14].

4.5 Combination of Momentum and Turbo f-ICA's

It is possible to use both momentum and turbo effects.

$$\tilde{\Delta}_f W(t) = \tilde{\Delta}W(t) + \mu_t \tilde{\Delta}W(t-\tau) + \nu_t \tilde{\Delta}\hat{W}(t+\tau) \tag{54}$$

We can give reasoning to use this update equation from the definitions of the convex divergence. Definition (1) means that the current environment is considered to be $q(y)$. On the other hand, definition (2) takes $p(y)$ as the current environment. Thus,

$$\begin{aligned} D(p\|q) &= D_{f_1}(p\|q) + D_{f_2}(q\|p) \\ &= D_{f_1}(p\|q) + D_{g_2}(p\|q) \end{aligned} \tag{55}$$

gives the momentum and turbo terms.

4.6 Speed Evaluation

4.6.1 Experimental Evaluation

Since we are given $\{x(t)\}_{t=1}^{N}$ as a mixture source vectors, the expectation $E[\cdot]$ is approximated by $\frac{1}{T}\sum_{i=1}^{T}[\cdot]$, where T is the number of samples in a selected window. The case of $T = N$ is the full batch mode. If we use $T < N$ as a window, the method becomes a semi-batch mode. If $T = 1$, the case is an incremental learning. It is possible to choose a window size smaller than N for the look ahead part so that computation is alleviated.

We chose mixtures of five time series as benchmarking problems. The non-linearity of $\varphi(y) = y^3$ [1] was used. The convergence speed was measured by the cross-talking error [5] which checks the closeness of the matrix WA to $\Lambda\Pi$.

4.6.2 Test 1: Comparison with the best ρ for the MMI

Consider a case of plain minimum-mutual information ICA. If we were a priori given the largest $\{\rho_t\}_{t\geq 1}$ which gives convergence of $\{W(t)\}_{t\geq 1}$, this case would give an almost optimally fast speed. But, such a series of constants $\{\rho_t\}_{t\geq 1}$ can never be given before learning iterations since these numbers depend on unknown sources and their probability densities. Estimation of ρ_t at each step, if possible, should be equivalently heavy to the source separation. Therefore, we have to fix $\rho_t = \rho$. Even so, a limiting large ρ can not be found in advance because this figure also depends on unknown probability densities. Our first experiment is (i) to obtain such a limiting ρ, (ii) then, to compare with the f-ICA. $\rho = 0.50$ was found to be the limit after 30 trial runs.

Table I Comparison of iteration counts for ICA's with limiting $\rho = 0.50$.

plain MMI	momentum	turbo	m+t
23	18	9	7

Thus, the f-ICA strategies are effective.

4.6.3 Test 2: Comparison using a rule-of-thumb ρ

We have a rule-of-thumb; $\rho = 0.1$. Table II compares this case.

Table II Comparison of iteration counts for ICA's with $\rho = 0.1$.

plain MMI	momentum	turbo	m+t
115	39	16	14

Recommended figures are $m_f = 0.7$ and $\bar{m}_f = 0.85$.

5 Applications to Brain fMRI Maps

5.1 Use of Source Data

The test data of the brain fMRI is a set of $S \times T$ two-dimensional images. Here, $S = 7$ is the number of slices of a human head. Each slice contains 128×128 pixels as a two-dimensional array. $T = 68$ is the number of samples in the time axis.

Three visual patterns were shown to a tested person. They are (r) a dark background with a small red cross at the center, (s) a still image with many white squares located randomly, and (m) two groups of moving squares in opposite directions each other. Our goal is to find (i) a set of independent maps, and (ii) its associated activation patterns which are relevant to recognizing the moving image.

For the experiment, we picked up a data set of "*ssmmssmm*." Therefore, $x(t)$, $(t = 1, \ldots, 128\times 128)$, is a set of eight-dimensional column vectors corresponding to "*ssmmssmm*." The index t stands for one of pixel positions in 128×128 region. Thus, $y(t)$, $(t = 1, \ldots, 128 \times 128)$, give eight images of extracted maps. Since

$$x(t) = W^{-1}y(t) \stackrel{\text{def}}{=} Uy(t), \tag{56}$$

the column vectors in the matrix U can be interpreted as activation patterns of the eight maps.

Fig. 1 is a found map corresponding to "low-low-high-high-low-low-high-high." This illustration is a superposition of active areas (black) and anatomical data (half tone). The experiment was successful by using $\varphi(y) = tanh(y)$. The black horizontal line at the one fourth rear of the section is a zero-frequency artifact. Thus, we found a very active region at the rear part of the right hemisphere for this tested person (a young male adult).

6 Concluding Remarks

A class of ICA algorithms that use the convex divergence as a surrogate function was derived. This was called the f-ICA. The minimization of the convex divergence was carried out by gradient descent methods. This lead to implementation as a momentum term and/or a look-ahead term. Resulting software implementations showed remarkable speed as a gradient descent type. This was because effective use of past and/or future data. Additional software complexity is light.

In the application part, we showed distillation of task-related fMRI brain maps. An active area corre-

sponding to the recognition of the moving images was found at the rear of the right hemisphere.

Acknowledgments

The authors are quite thankful to Dr. R. Allen Waggoner, Dr. Keiji Tanaka and Dr. Hiroshige Takeichi of RIKEN BRI for permitting them to try out the test data set.

The authors are also indebted to the Grant-in-Aid for Scientific Research as well as the High Technology and New Technology Projects of Waseda University.

References

[1] C. Jutten and J. Herault, Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing, vol. 24, pp. 1-20, 1991.

[2] J.-F. Cardoso and A. Souloumiac, Blind beamforming for non Gaussian signals, IEE Proceedings-F, vol. 140, No. 6, pp. 362-370, 1993.

[3] P. Comon, Independent component analysis, A new concept?, Signal Processing, vol. 36, pp. 287-314.

[4] A.J. Bell and T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[5] H.H. Yang and S. Amari, Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

[6] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks, vol. 10, pp. 626-634, 1999.

[7] Y. Matuyama, N. Katsumata, Y. Suzuki and S. Imahara, The α-ICA algorithm, Proc. ICA2000, pp. 297-302, 2000.

[8] Y. Matsuyama, T. Niimoto, N. Katsumata, Y. Suzuki and S. Furukawa, α-EM algorithm and α-ICA learning based upon extended logarithmic information measures, Proc. IJCNN2000, vol. III, pp. 351-356, 2000.

[9] Y. Matsuyama and S. Imahara, The α-ICA algorithm and brain map distillation from fMRI images, Proc. ICONIP2000, vol. 2, pp. 708-713, 2000.

[10] I. Csiszár, Information-type measures of difference of probability distributions and indirect observations, Studia Sci. Math. Hungarica, vol. 2, pp. 299-318, 1967.

[11] T.M. Cover and J.A. Thomas, *Elements of Information Theory,* Wiley, 1991.

[12] S. Amari, Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

[13] S. Amari T-P. Chen and A.J. Cichocki, Nonholonomic constraints in learning blind source separation, Proc. ICONIP'97, vol. 1, pp. 633-636, 1997.

[14] Y. Matsuyama and S. Imahara, Independent component analysis by convex divergence minimization: Applications to brain fMRI analysis, Proc. IJCNN2001, vol. 1, pp. 412-417, 2001.

[15] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski, Spacially independent activity patterns in functional MRI data during the Stroop color-naming task, Proc. Natl. Acad. Sci. USA, vol. 95, pp. 803-810, 1998.

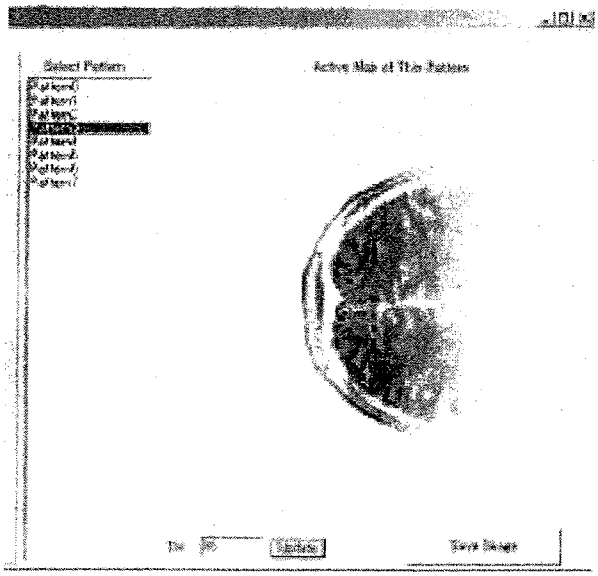


Figure 1: Activation pattern for "*ssmmssmm*" and its associated map.

3rd International Conference on Independent Component Analysis and Signal Separation

San Diego, California December 9-13, 2001

http://ica2001.org

ICA 2001 SAN DIEGO

Organizing Committee:

General Chair	Terrence Sejnowski
Program Chair	Te-Won Lee
Publication Chair	Tzyy-Ping Jung
Publicity Chair	Scott Makeig
Communications Chair	Javier Movellan
Business and Financial Services	John Staight
Host Institution	Institute for Neural Computation University of California, San Diego
Proceedings Editors:	Lee, Jung, Makeig, Sejnowski

CONVEX DIVERGENCE AS A SURROGATE FUNCTION FOR INDEPENDENCE: THE f-DIVERGENCE ICA

Yasuo Matsuyama, Naoto Katsumata and Shuichiro Imahara

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan
{yasuo, katsu, shoe16}@wizard.elec.waseda.ac.jp

ABSTRACT

The convex divergence is used as a surrogate function for obtaining independence of random variables described by the joint probability density. If the kernel convex function is twice continuously differentiable, this case reveals a class of generalized logarithm. This class of logarithms gives generalizations of the score function and the Fisher information matrix which are related to the Cramér-Rao bound. Guided by these properties, independent component analysis (ICA) using the convex divergence is presented. Obtained algorithms use the past and/or future data. Software implementation is easy and beats the minimum mutual information ICA in the speed. Real world experiments on brain fMRI are also performed.

1. INTRODUCTION

The convex divergence [7] measures difference of two probabilities by using a class of convex functions. By choosing the convex function appropriately, this measure is non-negative, and is zero if and only if two probability measures are equal almost everywhere.

Consider the case that one probability measure is a joint probability density and the other is a product probability density. Then, the convex divergence reflects a degree of independence inherent in the random variables described by the joint probability density. Thus, an iterated minimization of the convex divergence on the structural parameters of the joint probability density can be interpreted as a learning process towards source signal separation into independent components. Such an issue using the mutual information was addressed in [3], [5], [8], [15] and many others. In [9] and [12], a super class of the mutual information, i.e., the α-divergence, was used from an interest as a generalization. In this paper, we start with the aforementioned convex divergence from theoretical interests. It is found that the derived algorithms have a merit of speedup. The algorithms are simple and easily applicable to real world data. Early versions can be found in [10], [11].

Organization of this paper is as follows. Section 2 reviews fundamental properties of the convex divergence. It is newly found that the case of twice continuously differentiable convex functions brings about a class of generalized logarithms. Discussions therein also give generalizations of the score function and the Fisher information matrix. A relationship to Cramér-Rao inequality is also revealed. There, a scale factor is introduced. Section 3 gives gradient descent for the convex divergence and ICA. Section 4 shows methods of software implementations. Use of the past and/or future is considered. Speed evaluations by simulations are given. Applications to real world data such as brain fMRI are also addressed. Section 5 gives concluding remarks such that the α-divergence essentially "spans" the methods of the f-divergence if the convex function is twice continuous differentiable.

This work was partially supported by Grant-in-Aid for Scientific Research, and Waseda University's Research Projects on High Technologies and New Technologies.

2. CONVEX DIVERGENCE AND ITS PROPERTIES

2.1. Definitions and Basic Properties

The convex divergence between two probability densities p and q is defined by the following equations [7].

$$D_f(p\|q) = \int_{\mathcal{Y}} q(y) f(p(y)/q(y)) dy \quad (1)$$

$$= \int_{\mathcal{Y}} p(y) g(q(y)/p(y)) dy \quad (2)$$

$$= D_g(q\|p) \geq g(1) = f(1). \quad (3)$$

Here, $\mathcal{Y}$ is chosen to be a K-dimensional Euclidian space. The function $f(r)$, $r \in (0, \infty)$, is convex. The

dual function $g(r)$ satisfies

$$g(r) = rf(1/r) \tag{4}$$

so that it is also convex. The inequality (3) is the equality if and only if $p(y) = q(y)$, y-a.e. Since the normalization of $f(1)$ is arbitrary, we can choose

$$f(1) = g(1) = 0. \tag{5}$$

Then, the convex divergence can be regarded as a directed distance between p and q.

2.2. Convex Functions with Twice Continuous Differentiability

In the definitions of the convex divergences D_f and D_g, differentiability of $f(r)$ and $g(r)$ is not necessarily required. But, we are interested in the case that these functions are twice continuously differentiable. This is because we will derive learning algorithms based upon gradients. Then, for

$$C \stackrel{\text{def}}{=} \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} \in (-\infty, \infty), \tag{6}$$

we have the following equalities around $r = 1$.

$$\frac{f(r)}{f''(1)} = \frac{1}{C(1-C)}(r - r^C) + o(1) \tag{7}$$

$$\frac{g(r)}{g''(1)} = \frac{-1}{C(1-C)}(r^{1-C} - 1) + o(1) \tag{8}$$

Here, $o(1)$ is a higher order term. It is important to note that

$$\begin{aligned} \frac{1}{C(1-C)}(r - r^C) &= \left\{\frac{1}{C}r^C\right\}\left\{\frac{1}{1-C}(r^{1-C} - 1)\right\} \quad (9) \\ &\stackrel{\text{def}}{=} U^{(C)}(r)L^{(C)}(r) \quad (10) \end{aligned}$$

In the above expression,

$$L^{(C)}(r) = \frac{1}{1-C}(r^{1-C} - 1) \tag{11}$$

is a compelling function. This is a parameterized class of monotone functions whose convexity is controlled by the parameter C from the ultimate concavity to the ultimate convexity. It is important to note that

$$L^{(1)}(r) = \log r. \tag{12}$$

Thus, $L^{(C)}(r)$ can be regarded as a wide-sense logarithm. We can call this function the C-logarithm. If the argument r is replaced by a probability density p, $L^{(C)}(p)$ can be interpreted as a generalized score function.

2.3. Information Matrix and Cramér-Rao Bound

By using the C-logarithm, we have the following equality.

$$\begin{aligned} M^{(C)}(\varphi) &\stackrel{\text{def}}{=} E_p\left[Cp^{-2(1-C)}\left(\frac{\partial L_C}{\partial \varphi}\right)\left(\frac{\partial L_C}{\partial \varphi^T}\right)\right] \quad (13) \\ &= -E_p\left[p^{-(1-C)}\left(\frac{\partial^2 L_C}{\partial \varphi \partial \varphi^T}\right)\right] \quad (14) \end{aligned}$$

This equality can be regarded as a generalization of the Fisher information matrix. In fact, it holds that

$$M^{(C)}(\varphi) = CM^{(1)}(\varphi) = CF(\varphi). \tag{15}$$

Here, $F(\varphi)$ is the Fisher information matrix. The constant C can be regarded as a scale factor.

The information matrix $M^{(C)}(\varphi)$ is related to the Cramér-Rao bound. Let $h(\varphi)$ be an unknown vector function of a vector variable φ for a statistical model $p_{Y|\varphi}(y|\varphi)$. Let $\hat{h}(Y)$ be an unbiased estimate for $h(\varphi)$. Let

$$V(\hat{h}(Y)) \stackrel{\text{def}}{=} \left[\text{Cov}\left(\hat{h}_i(Y), \hat{h}_j(Y)\right)\right] \tag{16}$$

and

$$\Omega(\varphi) \stackrel{\text{def}}{=} \frac{\partial h(\varphi)}{\partial \varphi^T}. \tag{17}$$

Then, the following inequality holds.

$$\begin{aligned} V(\hat{h}(Y)) &\geq C\Omega(\varphi)\{M^{(C)}(\varphi)\}^{-1}\Omega(\varphi)^T \\ &= \Omega(\varphi)\{M^{(1)}(\varphi)\}^{-1}\Omega(\varphi)^T \quad (18) \end{aligned}$$

This corresponds to the Cramér-Rao inequality. Thus, the bound is not degraded by the choice of C.

3. CONVEX DIVERGENCE ICA

3.1. Gradient of the Convex Divergence

In the problem of ICA, we are given a set of N vector random variables.

$$x(n) = \text{col}[x_1(n), \cdots, x_K(n)], \quad (n = 1, \cdots, N). \tag{19}$$

Each $x(n)$ is a mixture by an unknown matrix A such that

$$As(n) = x(n). \tag{20}$$

Here, the vector

$$s(n) = \text{col}[s_1(n), \cdots, s_K(n)] \tag{21}$$

is also unknown except that its components are independent each other. Then, we want to find a demixing matrix $W = \Lambda\Pi A^{-1}$ so that the components of

$$Wx(n) \stackrel{\text{def}}{=} y(n) = \text{col}[y_1(n), \cdots, y_K(n)] \tag{22}$$

are independent each other for every n. Here, Λ is a nonsingular diagonal matrix and Π is a permutation matrix. Both matrices are also unknown.

Let

$$p(y) = p(y_1, \cdots, y_K) \tag{23}$$

be a joint probability density and

$$q(y) = \prod_{i=1}^{K} q_i(y_i) \tag{24}$$

be a product probability density. Then, we have

$$\begin{aligned} I_f(\bigwedge_{i=1}^{K} Y_i) &\stackrel{\text{def}}{=} D_f\left(p(y_1, \cdots, y_K) \| \prod_{i=1}^{K} q_i(y_i)\right) \\ &\stackrel{\text{def}}{=} D_f(p(y)\|q(y)) \\ &= D_g(q(y)\|p(y)) \\ &= I_g(\bigwedge_{i=1}^{K} Y_i) \\ &= \int_{\mathcal{X}} p(x) g\left(\frac{|W|q(y)}{p(x)}\right) dx. \end{aligned} \tag{25}$$

Here, we used

$$dy = |W|dx \tag{26}$$

as well as

$$p(y)dy = p(x)dx. \tag{27}$$

The symbol "$\bigwedge$" is used instead of ";" which appears in standard references [6]. It is important to observe that the determinant $|W|$ appears at only one place in the last expression of (25).

The negative gradient is obtained as follows.

$$\begin{aligned} -\nabla I_g(\bigwedge_{i=1}^{K} Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^{K} Y_i)}{\partial W} \\ &= \int_{\mathcal{X}} |W| q(y) g'\left(\frac{|W|q(y)}{p(x)}\right) \left\{W^{-T} - \varphi(y)x^T\right\} dx \\ &= -\nabla I_f(\bigwedge_{i=1}^{K} Y_i) \end{aligned} \tag{28}$$

Here,

$$g'(r) = \frac{d}{dr} g(r). \tag{29}$$

and

$$-\varphi(y) = \mathrm{col}\left[\frac{q_1'(y_1)}{q_1(y_1)}, \ldots, \frac{q_K'(y_K)}{q_K(y_K)}\right]. \tag{30}$$

Then, a simple update equation is

$$W(t+1) = W(t) + \Delta_f W(t) \tag{31}$$

with

$$\begin{aligned} \Delta_f W(t) &= \rho_t \left\{-\nabla I_f(\bigwedge_{i=1}^{K} Y_i)\right\}_{W=W(t)} \\ &= \rho_t \left\{-\nabla I_g(\bigwedge_{i=1}^{K} Y_i)\right\}_{W=W(t)} \\ &= \Delta_g W(t). \end{aligned} \tag{32}$$

Here, t is the index for iteration and ρ_t is a learning rate.

3.2. Removal of the Inverse Matrix

In Equation (28), a matrix inverse and transpose W^{-T} appears. The matrix inversion and transposition can be removed by using a natural or relative gradient [1], [4]. By considering (15), we multiply CW^TW. Then, we have

$$\begin{aligned} -\tilde{\nabla} I_g(\bigwedge_{i=1}^{K} Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^{K} Y_i)}{\partial W}(CW^TW) \\ &= -C \int_{\mathcal{X}} q(y) g'\left(\frac{|W|q(y)}{p(x)}\right) \left\{I - \varphi(y)x^T W^T\right\} |W| dx\, W \\ &= -C \int_{\mathcal{Y}} q(y) g'\left(\frac{q(y)}{p(y)}\right) \left\{I - \varphi(y)y^T\right\} dy\, W. \end{aligned} \tag{33}$$

An important next step is how to evaluate the core of the integrand of (33). It is a key to observe

$$qg'(q/p) = -g''(1)p + \{g'(1) + g''(1)\}q + o(1) \tag{34}$$

around $p \approx q$. Then, by virtue of (6), we have

$$\begin{aligned} q(y) g'(\tfrac{q(y)}{p(y)}) &= -g''(1)p(y)\left[1 + \frac{1 + \frac{g''(1)}{g'(1)}}{-\frac{g''(1)}{g'(1)}} \frac{q(y)}{p(y)}\right] + o(1) \\ &= -g''(1)p(y)\left[1 + \frac{1-C}{C}\frac{q(y)}{p(y)}\right] + o(1). \end{aligned} \tag{35}$$

Therefore, we have the following equation.

$$\begin{aligned} &-\frac{\partial I_f}{\partial W}(CW^TW) \\ &= -\frac{\partial I_g}{\partial W}(CW^TW) \\ &= f''(1)\Big[C\left\{I - E_{p(y)}[\varphi(y)y^T]\right\}W \\ &\quad + (1-C)\left\{I - E_{q(y)}[\varphi(y)y^T]\right\}W\Big] + o(1). \end{aligned} \tag{36}$$

Therefore,

$$0 < C \leq 1 \tag{37}$$

is a region of faster convergence with the ratio of $1 + (\frac{1-C}{C})\frac{q}{p}$. Note that $C = 1$ is the case of the minimum mutual information ICA because of (12).

3.3. Special Classes of the f-ICA

A useful class of convex functions satisfies the following equality.

$$f(xy) = kf(x)f(y) \tag{38}$$

The following function satisfies this equality.

$$f(r) = \frac{r^\beta}{k(\beta)} \tag{39}$$

Here, β and $k(\beta)$ should have a relationship so that $f(r)$ be a convex function. If we choose $f(1) = g(1) = 0$ and $f''(1) = g''(1) = 1$, then

$$f^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}), \tag{40}$$

and

$$g^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(1 - r^{\frac{1+\alpha}{2}}) \tag{41}$$

are such convex functions for $\alpha \in (-\infty, \infty)$. In this case,

$$C = \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} = \frac{1-\alpha}{2} \tag{42}$$

and

$$1 - C = 1 - \frac{f''(1)}{f'(1)} = 1 + \frac{g''(1)}{g'(1)} = \frac{1+\alpha}{2}. \tag{43}$$

Note that (37) corresponds to

$$-1 \leq \alpha < 1. \tag{44}$$

Thus, the α-divergence which uses $f^{(\alpha)}(r)$ and $g^{(\alpha)}(r)$ inherits the convexity control ability of the f-divergence through the parameter α instead of the parameter C.

4. IMPLEMENTATION OF THE CONVEX DIVERGENCE ICA

4.1. Non-Anticipatory Realization as the Momentum f-ICA

First, we observe that $q(y)$ is the target function of $p(y)$ such that

$$q(y) = \lim_{t\to\infty} p^{(t)}(y) \tag{45}$$

under an appropriate convergence criterion. Here, t is the index for the iteration. Then, there is a non-anticipatory approximation at the t-th iteration such that

$$q(y) \Leftarrow p^{(t)}(y) \quad \text{and} \quad p(y) \Leftarrow p^{(t-\tau)}(y). \tag{46}$$

By this approximation, we have the following sample-based learning algorithm.

[Momentum f-ICA]

If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta}W(t) + \mu_f \tilde{\Delta}W(t-\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \mu_f \{I - \varphi(y(t-\tau))y(t-\tau)^T\} W(t-\tau) \Big]\end{aligned} \tag{47}$$

Here,

$$\mu_f = \frac{C}{1-C} \tag{48}$$

Thus, we added a momentum term $\tilde{\Delta}W(t-\tau)$ weighted by μ_f. Figure 1 illustrates a flow of data and updates. Note that the case of $\mu_f = \frac{1-\alpha}{1+\alpha}$ corresponds to the α-ICA [9], [12]. Further special case of $\alpha = 1$, i.e., $\mu_f = 0$ is

$$\tilde{\Delta}_f W(t) = \tilde{\Delta}W(t) \tag{49}$$

which is the plain minimum mutual information method of [15].

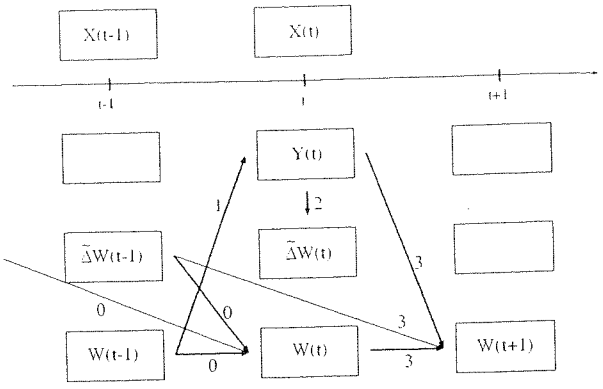


Figure 1: Diagram of the Momentum f-ICA.

4.2. Anticipatory Realization as the Turbo f-ICA

There is an anticipatory approximation at the t-th iteration such that

$$q(y) \Leftarrow p^{(t+\tau)}(y) \quad \text{and} \quad p(y) \Leftarrow p^{(t)}(y). \tag{50}$$

This is more natural than the momentum f-ICA since $p(y)$ has the present iteration index. Then, we have the following sample-based learning algorithm.

[Turbo (Look-ahead) f-ICA]

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta}W(t) + \nu_f \tilde{\Delta}W(t+\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \nu_f \{I - \varphi(\hat{y}(t+\tau))\hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \Big]\end{aligned} \tag{51}$$

Here,

$$\nu_f = \frac{1}{\mu_f} = \frac{1-C}{C} \tag{52}$$

The look-ahead terms $\hat{W}(t+\tau)$ and $\hat{y}(t+\tau)$ are estimations of $W(t+\tau)$ and $y(t+\tau)$ using the usual log-version. Thus, we added a predicted term $\tilde{\Delta}\hat{W}(t+\tau)$ weighted by ν_f. Figure 2 illustrates the flow of data and update terms. We comment here that there is a duality between Equations (47) and (51). We also note in advance that $\tau = 1$ works effectively enough for both causal and non-causal methods

4.3. Orthogonal f-ICA

Amari et al. [2] introduced an orthogonal ICA which is expected to suppress zero-power fake signals. The idea is to find an update term, say $\tilde{\Delta}^+ W$, which is orthogonal to $\tilde{\Delta}W$ so that

$$< \tilde{\Delta}W, \tilde{\Delta}^+ W >_W = 0. \tag{53}$$

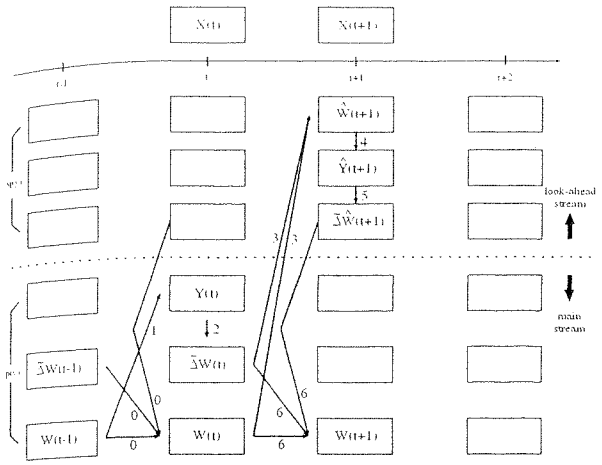


Figure 2: Diagram of the Momentum f-ICA.

Such an update term $\tilde{\Delta}^{+}W$ is obtained as follows. Let

$$\Lambda = \mathrm{diag}\,[\lambda_i]_{i=1}^{K} \tag{54}$$

be a non-singular diagonal matrix. Let

$$W + \tilde{\Delta}W = (I + d\Lambda)W. \tag{55}$$

Then, it holds that

$$\tilde{\Delta}^{+}W = \rho\{\Lambda - \varphi(y)y^{T}\}W, \tag{56}$$

where

$$\Lambda = \mathrm{diag}\,[\varphi_i(y_i)y_i]_{i=1}^{K}. \tag{57}$$

We can obtain four types of orthogonal f-ICA algorithms as is given in [10].

4.4. Combination of Momentum and Turbo f-ICA's

It is possible to use both momentum and turbo effects.

$$\tilde{\Delta}_f W(t) = \tilde{\Delta}W(t) + \mu_t \tilde{\Delta}W(t-\tau) + \nu_t \tilde{\Delta}\hat{W}(t+\tau) \tag{58}$$

We can give reasoning to use this update equation from the definitions of the convex divergence. Definition (1) means that the current environment is considered to be $q(y)$. On the other hand, definition (2) takes $p(y)$ as the current environment. Thus,

$$\begin{aligned} D(p\|q) &= D_{f_1}(p\|q) + D_{f_2}(q\|p) \\ &= D_{f_1}(p\|q) + D_{g_2}(p\|q) \end{aligned} \tag{59}$$

gives the joint momentum and turbo f-ICA.

4.5. Experiments

4.5.1. Experimental Evaluation

Since we are given $\{x(n)\}_{n=1}^{N}$ as a set of mixture source vectors, the expectation $E[\cdot]$ is approximated by $\frac{1}{T}\sum_{i=1}^{T}[\cdot]$ where T is the number of samples in a selected window. The case of $T = N$ is the full batch mode. If we use $T < N$ as a window, it becomes a semi-batch mode. If $T = 1$, the case is an incremental learning. It is possible to choose a window size smaller than N for the look-ahead part so that computation is alleviated.

We chose mixtures of five time series as benchmarking problems. The non-linearity of $\varphi(y) = y^3$ [8] was used. The convergence speed was measured by the cross-talking error [15] which checks the closeness of the matrix WA to $A\Pi$.

Our first experiment is, (i) to obtain a limit large ρ for the plain MMI. $\rho = 0.50$ was found to be the limit after many trial runs. Following to this step, (ii) the f-ICA was applied by using this ρ. Table I shows the speed of convergence.

Table I Iteration counts for ICA's with $\rho = 0.50$.

plain MMI	momentum	turbo	m+t
23	18	9	7

Thus, the f-ICA strategies are effective.

Next, we try experiments from a different angle. We have a rule-of-thumb; say, $\rho = 0.1$. Table II compares this case.

Table II Iteration counts for ICA's with $\rho = 0.1$.

plain MMI	momentum	turbo	m+t
115	39	16	14

Recommended figures are $C = 0.7$ for the momentum f-ICA, and $1 - C = 0.85$ for the turbo f-ICA .

4.5.2. Applications

After [14], we have tried processing of brain fMRI data. We applied the f-ICA to find active areas when a tested person watches moving images. Figure 3 shows an active area at the rear of the right hemisphere (male). Because of the f-ICA, a conventional personal computer was enough.

5. CONCLUDING REMARKS

The convex divergence, or the f-divergence, is an intriguing quantity which measures a directed distance of two probability densities. If the kernel function is convex with twice continuous differentiability, we can find an accompanied function which can be regarded as a generalized logarithm In this context, there are

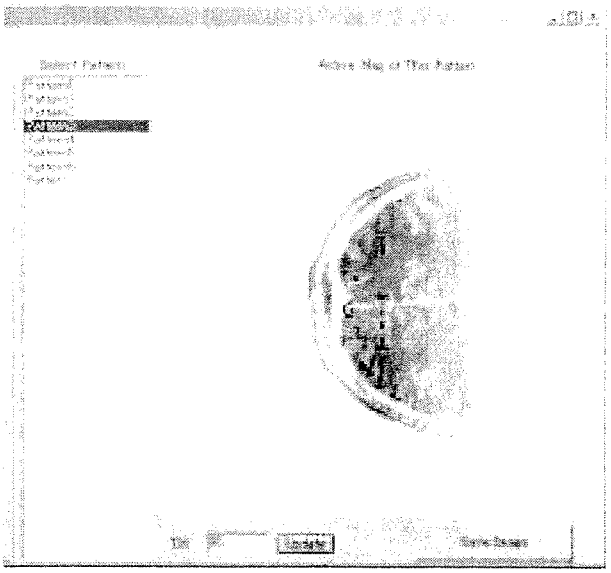


Figure 3: Activation pattern for "*ssmmssmm*" and its associated map.

generalizations of the score function and the information matrix. The Cramér-Rao bound remains non-deteriorated.

In this paper, we focused on the derivation of ICA algorithms using the f-divergence as a surrogate function for independence. It is worth noting that the EM algorithm was also extended by using a divergence measure [13]. The present paper revealed that the α-divergence used in [13] essentially "spans" the methods of the f-divergence.

ACKNOWLEDGEMENT

The authors are quite thankful to Dr. R. Allen Waggoner, Dr. Keiji Tanaka and Dr. Hiroshige Takeichi of RIKEN BRI for permitting them to try out the test data set.

REFERENCES

[1] S. Amari, Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

[2] S. Amari T-P. Chen and A.J. Cichocki, Non-holonomic constraints in learning blind source separation, Proc. ICONIP'97, vol. 1, pp. 633-636, 1997.

[3] A.J. Bell and T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[4] J.-F. Cardoso and B. H. Laheld, Equivariant adaptive source separation, IEEE Trans. Signal Processing, vol. 44, pp. 3017-3030, 1996.

[5] P. Comon, Independent component analysis, A new concept?, Signal Processing, vol. 36, pp. 287-314, 1994.

[6] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley, 1991.

[7] I. Csiszár, Information-type measures of difference of probability distributions and indirect observations, Studia Sci. Math. Hungarica, vol. 2, pp. 299-318, 1967.

[8] C. Jutten and J. Herault, Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing, vol. 24, pp. 1-20, 1991.

[9] Y. Matsuyama and S. Imahara, The α-ICA algorithm and brain map distillation from fMRI images, Proc. ICONIP2000, vol. 2, pp. 708-713, 2000.

[10] Y. Matsuyama and S. Imahara, Independent component analysis by convex divergence minimization: Applications to brain fMRI analysis, Proc. IJCNN2001, vol. x, pp. y-z, 2001.

[11] Y. Matsuyama, N. Katsumata and S. Imahara, Independent component analysis using convex divergence, Proc. ICONIP2001, vol. x, pp. y-z, 2001.

[12] Y. Matuyama, N. Katsumata, Y. Suzuki and S. Imahara, The α-ICA algorithm, Proc. ICA2000, pp. 297-302, 2000.

[13] Y. Matsuyama, T. Niimoto, N. Katsumata, Y. Suzuki and S. Furukawa, α-EM algorithm and α-ICA learning based upon extended logarithmic information measures, Proc. IJCNN2000, vol. III, pp. 351-356, 2000.

[14] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski, Spacially independent activity patterns in functional MRI data during the Stroop color-naming task, Proc. Natl. Acad. Sci. USA, vol. 95, pp. 803-810, 1998.

[15] H.H. Yang and S. Amari, Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

The 2002 IEEE World Congress ON COMPUTATIONAL INTELLIGENCE

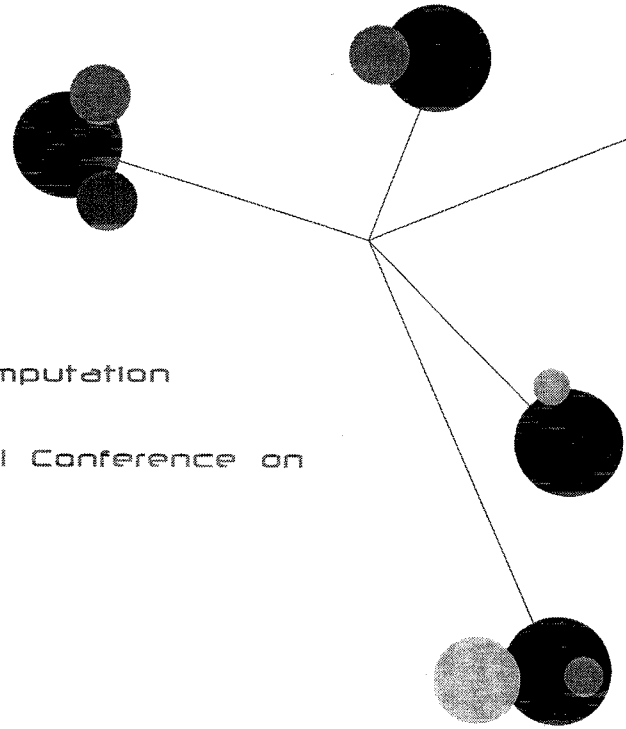
May 12 – 17, 2002
Hilton Hawaiian
Village Hotel
Honolulu,
Hawaii

Change Conferences

Congress on
Evolutionary Computation

IEEE International Conference on
Fuzzy Systems

2002
International Joint Conference on Neural Networks



OPTIMIZATION TRANSFER FOR COMPUTATIONAL LEARNING:

A Hierarchy from f-ICA and alpha-EM to their Offsprings

Yasuo Matsuyama †, Shuichiro Imahara ‡ and Naoto Katsumata †

† Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan
‡ Toshiba, Co., Japan
{yasuo, shoe16, katsu}@wizard.elec.waseda.ac.jp

Abstract - Likelihood optimization methods for learning algorithms are generalized and faster algorithms are provided. The idea is to transfer the optimization to a general class of convex divergences between two probability density functions. The first part explains why such optimization transfer is significant. The second part contains derivation of the generalized ICA (Independent Component Analysis). Experiments on brain fMRI maps are reported. The third part discusses this optimization transfer in the generalized EM algorithm (Expectation-Maximization). Hierarchical descendants to this algorithm such as vector quantization and self-organization are also explained.

I. INTRODUCTION

Most learning algorithms for computational intelligence make use of cost functions to be optimized. For the learning, a large amount of data is fed into computational learning mechanism. The performance is measured in terms of a probabilistic cost. In this paper, we consider a hierarchy of cost functions for learning algorithms. Discussed are general classes of ICA (Independent Component Analysis) and EM (Expectation-Maximization algorithm). These algorithms contain traditional ICA and EM as special cases. These algorithms show faster convergence.

The motivation of this paper comes from the following experiment on the measurement of a class of stochastic cost functions. Figure 1 shows a result of measuring two types of spectral distances on power-normalized speech signals. The horizontal axis specifies a distance measure $d_{\mathrm{ncm}}(f,g)^2$ based on the linear prediction. This is called the normalized causal distance. The vertical axis gives a value measured by the cepstral distance, $d_{\log}(f/\sigma_f^2, g/\sigma_g^2)$. These two distances are regarded as of the same class since they differ just by one parameter (the parameter 'α' which appears in later sections). Each plot shows how these two distances are different or similar. Although the power

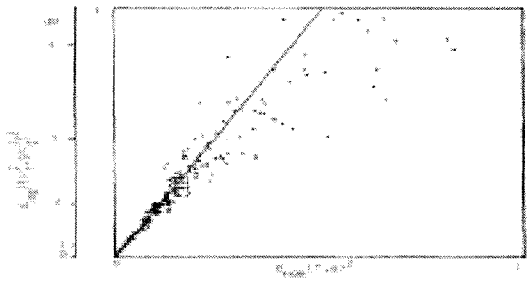


Figure 1: Scatter plots of two distances.

of two speech signals are normalized, the scatter plots still span a sector. Even in the region around the origin, this trend remains. Since zero by one distance ensures zero by the other, this figure gives us the following caution: A probabilistic cost function may not be simply approximated by another one.

Motivated by the above experiment, this paper shows that optimization transfer using a different cost function can be meritorious. The merits will be observed in learning speed and local optimality. Addressed learning algorithms include ICA (Independent Component Analysis) and EM algorithms (Expectation-Maximization). It is a further purpose to consider their algorithmic descendants. Experimental results are also given.

II. PRELIMINARIES

A. Convex Divergence

The convex divergence, or f-divergence, between two probability densities p and q is defined by the following

This work was partially supported by Grant-in-Aid for Scientific Research, and Waseda University's Research Projects on High Technologies and New Technologies.

0-7803-7278-6/02/$10.00 ©2002 IEEE

vol. 3, P. 1883 - 1888

equations [7].

$$\begin{aligned} D_f(p\|q) &= \int_{\mathcal{Y}} q(y) f(p(y)/q(y)) dy \\ &= \int_{\mathcal{Y}} p(y) g(q(y)/p(y)) dy \\ &= D_g(q\|p) \geq g(1) = f(1). \end{aligned} \tag{1}$$

Here, $\mathcal{Y}$ is chosen to be a K-dimensional Euclidian space. The function $f(r)$ is convex on $r \in (0, \infty)$. The dual function $g(r)$ is defined by

$$g(r) = rf(1/r), \tag{2}$$

which is also convex on $r \in (0, \infty)$. The inequality in (1) is the equality if and only if $p(y) = q(y)$, y-a.e. Since the normalization of $f(1)$ is arbitrary, we choose $f(1) = g(1) = 0$. Then, the convex divergence is regarded as a directed distance between p and q.

B. Convex Functions with Twice Continuous Differentiability

We are interested in the case that $f(r)$ is twice continuously differentiable on $r \in (0, \infty)$. This is because we will derive learning algorithms based upon gradients and/or closed-form updates. Then, for

$$c \stackrel{\text{def}}{=} \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} \in (-\infty, \infty), \tag{3}$$

we have the following equalities around $r = 1$.

$$\frac{f(r)}{f''(1)} = \frac{1}{c(1-c)}(r - r^c) + o(1) \tag{4}$$

$$\frac{g(r)}{g''(1)} = \frac{-1}{c(1-c)}(r^{1-c} - 1) + o(1) \tag{5}$$

Here, $o(1)$ is a higher order term. It is important to observe that

$$\begin{aligned} \frac{1}{c(1-c)}(r - r^c) &= \left\{\frac{1}{c} r^c\right\} \left\{\frac{1}{1-c}(r^{1-c} - 1)\right\} \qquad (6) \\ &\stackrel{\text{def}}{=} U^{(c)}(r) L^{(c)}(r). \qquad (7) \end{aligned}$$

In the above expression,

$$L^{(c)}(r) = \frac{1}{1-c}(r^{1-c} - 1) \tag{8}$$

is a compelling function. This is a parameterized class of monotone functions whose convexity is controlled by the parameter c from the ultimate concavity to the ultimate convexity. It is important to note that

$$L^{(1)}(r) = \log r. \tag{9}$$

Thus, $L^{(c)}(r)$ can be regarded as a wide-sense logarithm. We call this function the c-logarithm. Since the argument r is replaced by a probability density p, $L^{(c)}(p)$ is interpreted as a generalized score function.

C. A Special Class: The α-Divergence

A useful class of convex functions satisfies the following equality.

$$f(xy) = kf(x)f(y) \tag{10}$$

If we choose $f(1) = g(1) = 0$ and $f''(1) = g''(1) = 1$, then

$$f^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}) \tag{11}$$

and

$$g^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(1 - r^{\frac{1+\alpha}{2}}) \tag{12}$$

satisfy these requirements for $\alpha \in (-\infty, \infty)$. In this case,

$$c = \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} = \frac{1-\alpha}{2} \tag{13}$$

and

$$1 - c = 1 - \frac{f''(1)}{f'(1)} = 1 + \frac{g''(1)}{g'(1)} = \frac{1+\alpha}{2}. \tag{14}$$

Thus, the α-divergence which uses $f^{(\alpha)}(r)$ and $g^{(\alpha)}(r)$ inherits the convexity control ability of the f-divergence through the parameter α instead of the parameter c. This property is global since "$o(1)$" of (4) and (5) does not appear for $f^{(\alpha)}(r)$ and $g^{(\alpha)}(r)$.

D. Information Matrix and Cramér-Rao Bound

By using the c-logarithm, we have the following equality on information matrices.

$$\begin{aligned} M^{(c)}(\varphi) &\stackrel{\text{def}}{=} E_p\left[c\, p^{-2(1-c)} \left(\frac{\partial L_c}{\partial \varphi}\right) \left(\frac{\partial L_c}{\partial \varphi^T}\right)\right] \qquad (15) \\ &= -E_p\left[p^{-(1-c)} \left(\frac{\partial^2 L_c}{\partial \varphi \partial \varphi^T}\right)\right] \qquad (16) \end{aligned}$$

This equality can be regarded as a generalization of the Fisher information matrix. In fact, it holds that

$$M^{(c)}(\varphi) = cM^{(1)}(\varphi) = cF(\varphi). \tag{17}$$

Here, $F(\varphi)$ is the Fisher information matrix. The constant c can be regarded as a scale factor. We assume that the information matrices are positive definite, i.e.,

$$M^{(c)}(\varphi) > 0, \quad F(\varphi) > 0, \quad \text{and} \quad c > 0. \tag{18}$$

Then, the information matrix $M^{(c)}(\varphi)$ is related to the Cramér-Rao bound:

$$\begin{aligned} V(\hat{h}(Y)) &\geq c\,\Omega(\varphi)\{M^{(C)}(\varphi)\}^{-1}\Omega(\varphi)^T \\ &= \Omega(\varphi)\{M^{(1)}(\varphi)\}^{-1}\Omega(\varphi)^T \end{aligned} \tag{19}$$

Here, $\hat{h}(Y)$ is an unbiased estimate for $h(\varphi)$ which is an unknown vector function. The vector variable φ specifies a statistical model $p_{Y|\varphi}(y|\varphi)$ and

$$\Omega(\varphi) \stackrel{\text{def}}{=} \frac{\partial h(\varphi)}{\partial \varphi^T}. \tag{20}$$

0-7803-7278-6/02/$10.00 ©2002 IEEE

$V(\hat{h}(Y))$ is the covariance matrix of $\hat{h}(Y)$, i.e.,

$$V(\hat{h}(Y)) \stackrel{\text{def}}{=} \left[\text{Cov}\left(\hat{h}_i(Y), \hat{h}_j(Y)\right)\right]. \quad (21)$$

Equation (19) indicates that the bound is not degraded by the choice of c. Thus, the effect of choosing the cost function (1) appears in convergence properties, especially on the speed.

III. f-ICA

In this section, we apply the optimization transfer using the convex divergence to the ICA algorithm.

A. Gradient of the Convex Divergence

In the problem of ICA, we are given a set of vector random variables.

$$x(n) = \text{col}[x_1(n), \cdots, x_K(n)] = As(n), \ (n = 1, \cdots, N). \quad (22)$$

Here, the matrix A and the vector

$$s(n) = \text{col}[s_1(n), \cdots, s_K(n)] \quad (23)$$

are unknown except that

1. The components $s_i(n)$ and $s_j(n)$are independent each other $(i \neq j)$.
2. The components $s_i(n)$, $(i = 1, \cdots, K)$ are non-Gaussian except for at most one i.

Under such circumstance, we want to find a demixing matrix $W = \Lambda\Pi A^{-1}$ so that the components of

$$Wx(n) \stackrel{\text{def}}{=} y(n) = \text{col}[y_1(n), \cdots, y_K(n)] \quad (24)$$

are independent each other for every n. Here, Λ is a nonsingular diagonal matrix and Π is a permutation matrix, both of which are also unknown.

Let $p(y) = p(y_1, \cdots, y_K)$ be a joint probability density, and $q(y) = \prod_{i=1}^K q_i(y_i)$ be a product probability density. Then, the independence is obtained by the minimization of the following cost.

$$\begin{aligned} I_f(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} D_f\left(p(y_1, \cdots, y_K) \| \prod_{i=1}^K q_i(y_i)\right) \\ &\stackrel{\text{def}}{=} D_f(p(y) \| q(y)) \\ &= D_g(q(y) \| p(y)) \\ &= I_g(\bigwedge_{i=1}^K Y_i) \\ &= \int_{\mathcal{X}} p(x) g\left(\frac{|W| q(y)}{p(x)}\right) dx. \end{aligned} \quad (25)$$

The symbol "$\bigwedge$" is used instead of ";" which appears in standard references [6]. It is important to observe that the determinant $|W|$ appears at only one place in the last line of (25).

The negative gradient is obtained as follows:

$$\begin{aligned} -\nabla I_g(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W} \\ &= \int_{\mathcal{X}} |W| q(y) g'\left(\frac{|W| q(y)}{p(x)}\right) \left\{W^{-T} - \varphi(y) x^T\right\} dx \\ &= -\nabla I_f(\bigwedge_{i=1}^K Y_i), \end{aligned} \quad (26)$$

where

$$-\varphi(y) = \text{col}\left[\frac{q_1'(y_1)}{q_1(y_1)}, \ldots, \frac{q_K'(y_K)}{q_K(y_K)}\right]. \quad (27)$$

If we simply use

$$\Delta_f W(t) = \rho_t \left\{-\nabla I_f(\bigwedge_{i=1}^K Y_i)\right\}_{W=W(t)}, \quad (28)$$

the matrix inverse and transpose W^{-T} remains. This W^{-T} can be removed by using a natural or relative gradient [1], [4]. By considering (17), we multiply cW^TW. Then, we have [1]

$$\begin{aligned} -\tilde{\nabla} I_g(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W}(cW^TW) \\ &= -c \int_{\mathcal{X}} q(y) g'\left(\frac{|W| q(y)}{p(x)}\right) \left\{I - \varphi(y) x^T W^T\right\} |W| dx\, W \\ &= -c \int_{\mathcal{Y}} q(y) g'\left(\frac{q(y)}{p(y)}\right) \left\{I - \varphi(y) y^T\right\} dy\, W. \end{aligned} \quad (29)$$

B. Expansion for Concrete Algorithms

An important next step is how to evaluate the core of the integrand of (29) which contains $g'(r)$. Since we are not given a specific form of $g'(r)$, we use an expansion around $p \approx q$.

$$qg'(q/p) = -g''(1)p + \{g'(1) + g''(1)\}q + o(1) \quad (30)$$

Then, we have the following equations:

$$\begin{aligned} &-\frac{\partial I_f}{\partial W}(cW^TW) \\ &= -\frac{\partial I_g}{\partial W}(cW^TW) \\ &= f''(1)\Big[c\left\{I - E_{p(y)}[\varphi(y)y^T]\right\} W \\ &\quad + (1-c)\left\{I - E_{q(y)}[\varphi(y)y^T]\right\} W\Big] + o(1), \end{aligned} \quad (31)$$

and

$$\tilde{\Delta}_f W = -\rho_t \frac{\partial I_f}{\partial W} W^T W \quad (32)$$

Here, ρ_t is a small positive number called the learning rate. Thus,

$$0 < c \leq 1 \quad (33)$$

is a region for faster convergence with the ratio of $1 + (\frac{1-c}{c})\frac{q}{p}$. Note that $c = 1$ is the case of the minimum mutual information ICA because of (9).

[1] The constant c will be absorbed in the learning rate ρ_t in the final form of the increment. But, we explicitly specify this constant because of (17).

0-7803-7278-6/02/$10.00 ©2002 IEEE

IV. IMPLEMANTATION OF THE f-ICA

A. Non-Anticipatory Realization as the Momentum f-ICA

Since $p(y)$ and its target $q(y)$ appears in Equation (31) software implementation requires interpretations of $p(y)$ and $q(y)$. First, we consider a non-anticipatory approximation at the t-th iteration such that

$$p(y) \Leftarrow p^{(t-\tau)}(y) \quad \text{and} \quad q(y) \Leftarrow p^{(t)}(y). \tag{34}$$

By this method, we have the following sample-based learning algorithm [2].

[Momentum f-ICA]

If we use $p(y)$ as $p^{(t-\tau)}(y)$ and $q(y)$ as $p^{(t)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta} W(t) + \mu_f \tilde{\Delta} W(t-\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t)) y(t)^T\} W(t) \\ &\quad + \mu_f \{I - \varphi(y(t-\tau)) y(t-\tau)^T\} W(t-\tau) \Big]\end{aligned} \tag{35}$$

Here, $\mu_f = \frac{c}{1-c}$. Thus, we added a momentum term $\tilde{\Delta} W(t-\tau)$ weighted by μ_f. Note that the case of $\mu_f = \frac{1-\alpha}{1+\alpha}$ corresponds to the α-ICA [12], [16]. Further special case of $\alpha = 1$, i.e., $\mu_f = 0$ is

$$\tilde{\Delta}_f W(t) = \tilde{\Delta} W(t), \tag{36}$$

which is the plain minimum mutual information method of [3], [19].

B. Anticipatory Realization as the Turbo f-ICA

There is an anticipatory approximation at the t-th iteration such that

$$p(y) \Leftarrow p^{(t)}(y) \quad \text{and} \quad q(y) \Leftarrow p^{(t+\tau)}(y). \tag{37}$$

This is conceptually more natural than the momentum f-ICA since $p(y)$ has the present iteration index. Then, we have the following sample-based learning algorithm.

[Turbo (Look-ahead) f-ICA]

$$\begin{aligned}\tilde{\Delta}_f W(t) &= \tilde{\Delta} W(t) + \nu_f \tilde{\Delta} W(t+\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t)) y(t)^T\} W(t) \\ &\quad + \nu_f \{I - \varphi(\hat{y}(t+\tau)) \hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \Big]\end{aligned} \tag{38}$$

[2] "Sample-based" does not mean that the method is incremental. As will be observed later, the case of batch and/or semi-batch combination shows remarkable performance in speed improvement.

Here, $\nu_f = \frac{1}{\mu_f} = \frac{1-c}{c}$.

The look-ahead terms $\hat{W}(t+\tau)$ and $\hat{y}(t+\tau)$ are estimations of $W(t+\tau)$ and $y(t+\tau)$ using the usual log-version. Thus, we added a predicted term $\tilde{\Delta}\hat{W}(t+\tau)$ weighted by ν_f. We comment here that there is a duality between Equations (35) and (38). We also note in advance that $\tau = 1$ works effectively enough for both anticipatory and non-anticipatory methods.

C. Orthogonal f-ICA

Amari, Chen and Cichocki [2] introduced an orthogonal ICA which is expected to suppress zero-power fake signals. The idea is to find an update term, say $\tilde{\Delta}^+ W$, which is orthogonal to $\tilde{\Delta} W$ so that

$$< \tilde{\Delta} W, \tilde{\Delta}^+ W >_W = 0. \tag{39}$$

Such an update term $\tilde{\Delta}^+ W$ is obtained as follows. Let $\Lambda = \text{diag}\,[\lambda_i]_{i=1}^K$ be a non-singular diagonal matrix. Let

$$W + \tilde{\Delta} W = (I + d\Lambda) W. \tag{40}$$

Then, it holds that

$$\tilde{\Delta}^+ W = \rho \{\Lambda - \varphi(y) y^T\} W, \tag{41}$$

where $\Lambda = \text{diag}\,[\varphi_i(y_i) y_i]_{i=1}^K$. We can obtain four types of orthogonal f-ICA algorithms similarly to [13].

D. Combination of Momentum and Turbo f-ICA's

It is possible to use both momentum and turbo effects.

$$\tilde{\Delta}_f W(t) = \tilde{\Delta} W(t) + \mu_t \tilde{\Delta} W(t-\tau) + \nu_t \tilde{\Delta} \hat{W}(t+\tau) \tag{42}$$

We can give reasoning to use this update equation from the definitions of the convex divergence:

$$\begin{aligned} D(p\|q) &= D_{f_1}(p\|q) + D_{f_2}(q\|p) \\ &= D_{f_1}(p\|q) + D_{g_2}(p\|q). \end{aligned} \tag{43}$$

E. Experiments

1) Evaluation through Simulations: Since we are given $\{x(n)\}_{n=1}^N$ as a set of mixture source vectors, the expectation $E[\cdot]$ is approximated by $\frac{1}{T}\sum_{i=1}^T [\cdot]$ where T is the number of samples in a selected window. The case of $T = N$ is the full batch mode. If we use $T < N$ as a window, it becomes a semi-batch mode. If $T = 1$, the case is an incremental learning. It is possible to choose a window size smaller than N for the look-ahead part so that computation is alleviated.

We chose mixtures of five time series as benchmarking problems. The non-linearity of $\varphi(y) = y^3$ [9] was

0-7803-7278-6/02/$10.00 ©2002 IEEE

used. The convergence speed was measured by the cross-talking error [19] which checks the closeness of the matrix WA to $\Lambda\Pi$.

Our experiment is (i) to obtain a limit large ρ for the plain minimum-mutual-information ICA [3], [19]. By many trial runs, $\rho = 0.50$ was found to be the limit for the convergence. Then, the presented methods were tested by using this figure. Also tried is the following experiment: (ii) Since $\rho = 0.1$ is a usual number for a rule-of-thumb, we test this case too. Table I shows the speed of convergence.

Table I Iteration counts for ICA's.

ρ	plain MMI	momentum	turbo	m+t
0.5	23	18	9	7
0.1	115	39	16	14

Thus, the f-ICA strategies are effective. Recommended figures are $c = 0.7$ for the momentum f-ICA, and $1-c = 0.85$ for the turbo f-ICA .

Besides the result of Table I, it is necessary to consider practical execution speed. The following is the recommendation.

1. Pure momentum method is always recommended since its computational increase is only fractional.
2. Pure turbo method also shows good performance. But, the CPU performance is sometimes inferior to the momentum method due to the load at each iteration.
3. When the joint momentum and turbo method is used, it is recommended to reduce the window size of the turbo (e.g., to the half).

2) Application to Brain fMRI Maps: After [18], we have tried processing of brain fMRI data. We applied the f-ICA to find active areas when a tested person (adult male) pays attention to moving images. Figure 2 shows an active area at the rear of the right hemisphere (dorsal occipital cortex). Because of the f-ICA, a conventional personal computer was enough to perform this experiment.

V. EM ALGORITHM USING DIVERGENCE MEASURES

The EM algorithm (Expectation-Maximization) [8] is also possible to extend by the optimization transfer. In this case, the α-divergence by (11) and the α-logarithm (8) with (13) are used.

In the problem of EM algorithm, observed data, say y, is considered to be incomplete. There is assumed to be a complete-data x such that $y = y(x)$. We denote the complete-data pdf and incomplete-data pdf by

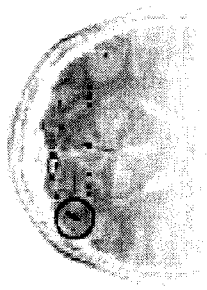


Figure 2: A Brain fMRI map.

$p_{X|\psi}(x|\psi)$ and $p_{Y|\psi}(y|\psi)$, respectively. Here, $\psi \in \Psi$ is a generic parameter for the probability density function. In the EM algorithm, we want to find

$$\psi^* = \arg\max_{\psi\in\Psi} p_{Y|\psi}(y|\psi) \tag{44}$$

In the case of the α-EM algorithm, this maximization is transferred to that of the conditional expectation. This is expressed by the following equality which is obtained from computing the α-divergence $D^{(\alpha)}(\varphi\|\psi)$ betweed $p_{X|Y,\varphi}(x|y,\varphi)$ and $p_{X|Y,\psi}(x|y,\psi)$:

$$\begin{aligned} L_Y^{(\alpha)}(\psi|\varphi) &= Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) \\ &\quad + \tfrac{1-\alpha}{2}\left\{\tfrac{p_{Y|\psi}}{p_{Y|\varphi}}\right\}^{\frac{1+\alpha}{2}} D^{(\alpha)}(\varphi\|\psi) \end{aligned} \tag{45}$$

Here, $L^{(\alpha)}$ is the α-logarithm, i.e., the c-logarithm with (13). The $Q^{(\alpha)}$-function is

$$Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) = E_{p_{X|Y,\varphi}}\left[L_X^{(\alpha)}(\psi|\varphi)\right]. \tag{46}$$

$L_X^{(\alpha)}(\psi|\varphi)$ is the α-log likelihood ratio of $p_{X|\psi}$ and $p_{X|\varphi}$ using $L^{(\alpha)}$. $L_Y^{(\alpha)}(\psi|\varphi)$ is the α-log likelihood ratio of incomplete-data pdfs. Thus, from the viewpoint of the optimization transfer, the maximization (44) is performed by that of $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$. Because of space limitation, computational issues of this $Q^{(\alpha)}$ is omitted [17].

It is important to note the following before closing this section. A hierarchy of various algorithms starting from the α-EM exists. They are soft and crisp vector quantization, self-organizing feature maps [11], which has relationships to encoding of emotion [10]. Further result will be given in the presentation.

VI. CONCLUDING REMARKS

In this paper, generalized learning algorithms obtained from the optimization transfer were discussed. This transferred optimization was based upon the convex divergence. There were discussed two classes of learning algorithms; the f-ICA and the α-EM. Main space of this paper were used on the f-ICA including the experiments on brain fMRI. It is the claim of this paper that the optimization transfer is a widely applicable method. In many cases, the optimization transfer using the extended logarithm shows faster convergence.

ACKNOWLEDGMENT

The authors are thankful to Dr. R. Allen Waggoner, Dr. Keiji Tanaka and Dr. Hiroshige Takeichi of RIKEN BRI for permitting them to try out the test data set.

REFERENCES

[1] S. Amari, Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

[2] S. Amari T-P. Chen and A.J. Cichocki, Nonholonomic constraints in learning blind source separation, Proc. ICONIP'97, vol. 1, pp. 633-636, 1997.

[3] A.J. Bell and T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[4] J.-F. Cardoso and B. H. Laheld, Equivariant adaptive source separation, IEEE Trans. Signal Processing, vol. 44, pp. 3017-3030, 1996.

[5] P. Comon, Independent component analysis, A new concept?, Signal Processing, vol. 36, pp. 287-314, 1994.

[6] T.M. Cover and J.A. Thomas, *Elements of Information Theory,* Wiley, 1991.

[7] I. Csiszár, Information-type measures of difference of probability distributions and indirect observations, Studia Sci. Math. Hungarica, vol. 2, pp. 299-318, 1967.

[8] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (woth discussion), J. R. Stat. Soc. B, vol. 39, pp.1-38, 1977.

[9] C. Jutten and J. Herault, Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing, vol. 24, pp. 1-20, 1991.

[10] Y. Matsuyama, Multiple descent cost competition: Restorable self-organization and multimedia image processing, IEEE Trans. Neural Networks, vol. 9, pp. 106-122, 1998.

[11] Y. Matsuyama, N. Takeda, S. Furukawa and T. Niimoto, A hierarchy from α-EM algorithm to vector quantization and self-organization, Proc. ICONIP, vol. 1,pp. 233-238, 1998.

[12] Y. Matsuyama and S. Imahara, The α-ICA algorithm and brain map distillation from fMRI images, Proc. ICONIP2000, vol. 2, pp. 708-713, 2000.

[13] Y. Matsuyama and S. Imahara, Independent component analysis by convex divergence minimization: Applications to brain fMRI analysis, Proc. IJCNN2001, vol. 1, pp. 412-417, 2001.

[14] Y. Matsuyama and N. Katsumata, Conex divergence as a surrogate function for independence: The f-divergence ICA, Proc ICA2001, vol. x, pp. y-z, 2001.

[15] Y. Matsuyama, N. Katsumata and S. Imahara, Independent component analysis using convex divergence, Proc. ICONIP2001, vol. 3, pp. 1173-1178, 2001.

[16] Y. Matuyama, N. Katsumata, Y. Suzuki and S. Imahara, The α-ICA algorithm, Proc. ICA2000, pp. 297-302, 2000.

[17] Y. Matsuyama, T. Niimoto, N. Katsumata, Y. Suzuki and S. Furukawa, α-EM algorithm and α-ICA learning based upon extended logarithmic information measures, Proc. IJCNN2000, vol. III, pp. 351-356, 2000.

[18] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski, Spacially independent activity patterns in functional MRI data during the Stroop color-naming task, Proc. Natl. Acad. Sci. USA, vol. 95, pp. 803-810, 1998.

[19] H.H. Yang and S. Amari, Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

ISITA NOLTA 2002

International Symposium on Information Theory and Its Applicatio

Xi'an International Conference Center, Xi'an, People's Republic of China
October 7-11, 2002.

ISITA 2002

ISAITA 2002 CONTENTS

Organized by :
The Society of Information Theory and Its Applicatio
Xi'an Jiaotong University

With the Technical co-sponsorship of:
The IEEE Information Theory Society
The Institution of Electronics, Information and Communication Engineers (IEICE)

ICF

Sponsored by :
International Communications Foundation
The Telecommunications Advancement Foundatic

International Symposium on Information Theory and Its Applications
Xi'an, PRC, October 7–11, 2002

Optimization Transfer Using Convex Divergence: f-ICA and alpha-EM Algorithm with Examples

Yasuo MATSUYAMA, Naoto KATSUMATA and Ryo KAWAMURA

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo, 169-8555, Japan
yasuo2@waseda.jp, {katsu, ryo}@wizard.elec.waseda.ac.jp

Abstract

Likelihood optimization using the convex divergence is discussed. This class of problems is a generalization of the log-likelihood optimization. First, basic properties including the logarithmic case, such as the information matrix, are discussed. Then, two types of problems are addressed. One is the independent component analysis. The separation of unknown independent components is achieved by minimizing the convex divergence. This is called the f-ICA. The other is the alpha-EM algorithm which use a "spanning" subclass of the convex divergence. In both problems, faster speed is obtained. Finally, a set of human brain's activation patterns is reported using the f-ICA.

1. Introduction

Likelihood optimization is a popular method in statistical and probabilistic information processing. Physical entities under this optimization can be versatile. Thus, the likelihood optimization has a wide variety of applications. In this paper, a general class of likelihood optimizations using the convex divergence, or the f-divergence [1], is presented. The methods given in this paper includes the log-likelihood optimization as a special case. That is, the optimization is transferred to the convex divergence. Basic properties concerning to this optimization transfer are discussed first.

Next, two main problems are discussed. The first one is the independent component analysis (ICA) using the f-divergence (f-ICA). The second one is the alpha-EM algorithm which uses the "spanning" subclass of the f-divergence. In both problems, faster speed is observed. Finally, a concrete example from the real world, the brain functional Magnetic Resonance Imaging (fMRI) is given using the f-ICA.

This work was supported by the Grant-in-Aid for Scientific Research, and High/New Technology Research Grants of Waseda University.

2. Preliminaries

2.1. Convex Divergence

The convex divergence [1] measures a directed distance between two probability densities p and q by using an adjustable convexity (or concavity).

$$\begin{aligned} D_f(p\|q) &= \int_{\mathcal{Y}} q(y) f(p(y)/q(y)) dy & (1)\\ &= \int_{\mathcal{Y}} p(y) g(q(y)/p(y)) dy & (2)\\ &= D_g(q\|p) \geq g(1) = f(1) \stackrel{\text{def}}{=} 0. & (3)\end{aligned}$$

Here, $f(r)$ is convex on $r \in (0, \infty)$, and $g(r) = rf(1/r)$. We are interested in the case that this $f(r)$ is twice continuously differentiable. Define

$$c \stackrel{\text{def}}{=} \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g(1)} \in (-\infty, \infty). \quad (4)$$

Then, the following expansion holds around $r = 1$.

$$\begin{aligned} \frac{f(r)}{f''(1)} &= \frac{1}{c(1-c)}(r - r^c) + o(1)\\ &= \left\{\frac{1}{c} r^c\right\} \left\{\frac{1}{1-c}(r^{1-c} - 1)\right\} + o(1). \quad (5)\end{aligned}$$

Here, $o(1)$ is the higher order term. From Equation (5), we find that

$$L^{(c)}(r) = \frac{1}{1-c}(r^{1-c} - 1) \quad (6)$$

is regarded as an extended class of the logarithm. In fact, $L^{(1)}(r) = \log r$ in the limit. This "c-logarithm" has relationships to the Fisher information matrix and the Cramér-Rao bound. Let L_c be an abbreviated notation of $L^{(c)}(p)$, where p stands for the probability density $p(y|\varphi)$. Then, we have

$$\begin{aligned} M^{(c)}(\varphi) &\stackrel{\text{def}}{=} E_p\left[c p^{-2(1-c)} \left(\frac{\partial L_c}{\partial \varphi}\right)\left(\frac{\partial L_c}{\partial \varphi^T}\right)\right] & (7)\\ &= -E_p\left[p^{-(1-c)} \left(\frac{\partial^2 L_c}{\partial \varphi \partial \varphi^T}\right)\right]. & (8)\end{aligned}$$

The case of $c = 1$ is reduced to the Fisher information matrix $F(\varphi)$.

$$M^{(c)}(\varphi) = cM^{(1)}(\varphi) = cF(\varphi). \quad (9)$$

Because of Equations (7) and (8), the usage of the information matrix $M^{(c)}(\varphi)$ does not deteriorate the Cramér-Rao bound [2]. We assume that underlying problems are not pathological, i.e.,

$$M^{(c)}(\varphi) > 0, \quad F(\varphi) > 0, \quad \text{and} \quad c > 0. \tag{10}$$

There is an important subclass of the convex function $f(r)$. The function

$$f^{(\alpha)}(r) = \frac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}) \tag{11}$$

generates the α-divergence [3], [4], [5]. In this case, Equation (5) holds without "$o(1)$" and

$$L^{(\alpha)}(r) = \frac{2}{1+\alpha}(r^{\frac{1+\alpha}{2}} - 1). \tag{12}$$

2.2. Optimization Transfer

In the following sections, we try optimizations of the likelihood functions related to the convex divergence. The independent component analysis is performed by *minimizing* the f-divergence (1) between the observed joint probability density p and the independent probability density q. On the other hand, the alpha EM algorithm *maximizes* the likelihood ratio function (12).

3. The f-ICA Algorithm

3.1. Derivation of the Algorithm

In the independent component analysis, the f-divergence is minimized as a measurement of the degree of independence. Traditional methods [6], [7], [8] minimize the mutual information or maximize the differential entropy.

Consider the following generalization of the mutual information.

$$I_f(\bigwedge_{i=1}^K Y_i) \stackrel{\text{def}}{=} D_f\left(p(y_1, \cdots, y_K) \| \prod_{i=1}^K q_i(y_i)\right) \tag{13}$$

This quantity measures how the joint probability density $p(y_1, \cdots, y_K)$ is close to that of the independence $\prod_{i=1}^K q_i(y_i)$.

In the problem of ICA, we are given a set of vector random variables.

$$x(n) = \mathrm{col}[x_1(n), \cdots, x_K(n)] = As(n), \ (n = 1, \cdots, N). \tag{14}$$

Here, the matrix A and the vector

$$s(n) = \mathrm{col}[s_1(n), \cdots, s_K(n)] \tag{15}$$

are unknown but the following: (i) The components $s_i(n)$ and $s_j(n)$ are independent each other $(i \neq j)$. (ii) The components $s_i(n)$, $(i = 1, \cdots, K)$, are non-Gaussian except for at most one i.

Under such conditions, we want to find a demixing matrix

$$W = \Lambda \Pi A^{-1} \tag{16}$$

so that the components of

$$Wx(n) \stackrel{\text{def}}{=} y(n) = \mathrm{col}[y_1(n), \cdots, y_K(n)] \tag{17}$$

are independent each other for every n. Here, Λ is a nonsingular diagonal matrix and Π is a permutation matrix. Both are unknown too.

For the estimation of the demixing matrix W, we use a gradient descent. In this case, we obtain

$$\begin{aligned} &-\nabla I_f(\textstyle\bigwedge_{i=1}^K Y_i) \stackrel{\text{def}}{=} -\frac{\partial I_f(\bigwedge_{i=1}^K Y_i)}{\partial W} \\ &= \int_{\mathcal{X}} |W| q(y) g'\left(\frac{|W| q(y)}{p(x)}\right) \left\{W^{-T} - \varphi(y) x^T\right\} dx. \end{aligned} \tag{18}$$

Here,

$$\varphi(y) = -\mathrm{col}\left[\frac{q_1'(y_1)}{q_1(y_1)}, \ldots, \frac{q_K'(y_K)}{q_K(y_K)}\right] \tag{19}$$

is a nonlinear function such as y^3 or $\tanh(y)$. For the natural or relative gradient [9], [10], we multiply cW^TW.

$$\begin{aligned} &-\tilde{\nabla} I_f(\textstyle\bigwedge_{i=1}^K Y_i) \stackrel{\text{def}}{=} -\frac{\partial I_f(\bigwedge_{i=1}^K Y_i)}{\partial W}(cW^TW) \\ &= -c \int_{\mathcal{X}} q(y) g'\left(\frac{|W| q(y)}{p(x)}\right) \left\{I - \varphi(y) x^T W^T\right\} |W| dx\, W \\ &= -c \int_{\mathcal{Y}} q(y) g'\left(\frac{q(y)}{p(y)}\right) \left\{I - \varphi(y) y^T\right\} dy\, W \\ &= f''(1)\Big[c\left\{I - E_{p(y)}[\varphi(y) y^T]\right\} W \\ &\qquad + (1-c)\left\{I - E_{q(y)}[\varphi(y) y^T]\right\} W\Big] + o(1). \end{aligned} \tag{20}$$

Here, the last equality is obtained by the expansion around $p \approx q$. Thus,

$$W(t+1) = W(t) + \tilde{\Delta}_f W(t), \tag{21}$$

with

$$\tilde{\Delta}_f W(t) = \rho(t)\left\{-\tilde{\nabla} I_f(\textstyle\bigwedge_{i=1}^K Y_i)\right\}_{W=W(t)}. \tag{22}$$

Here, $\rho(t)$ is a small positive number called the learning rate. We call the learning algorithm (21) the f-ICA. It is worth noting that $0 < c <= 1$ is a region of faster convergence with the ratio of $1 + \frac{1-c}{c}\frac{q}{p}$.

3.2. Software Realization

For the realization of the f-ICA algorithm as computer software, we consider the following items: (i)

Since we are given only sample observations, the expectation is approximated by repeated applications of given data. (ii) Since the probability density q is unknown, it is approximated by the time-shifted version of the current probability density function p.

Then, we have the following sample-based algorithms.

[Momentum f-ICA]

If we use $p(y)$ as $p^{(t-\tau)}(y)$ and $q(y)$ as $p^{(t)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned}\tilde{\Delta}_f W(t) &\stackrel{\text{def}}{=} \tilde{\Delta}W(t) + \mu_f \tilde{\Delta}W(t-\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \mu_f \{I - \varphi(y(t-\tau))y(t-\tau)^T\} W(t-\tau) \Big] \quad (23)\end{aligned}$$

Here, $\mu_f = \frac{c}{1-c}$. Thus, we added a momentum term $\tilde{\Delta}W(t-\tau)$ weighted by μ_f.

[Turbo (Look-ahead) f-ICA]

$$\begin{aligned}\tilde{\Delta}_f W(t) &\stackrel{\text{def}}{=} \tilde{\Delta}W(t) + \nu_f \tilde{\Delta}W(t+\tau) \\ &= \rho_t \Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \nu_f \{I - \varphi(\hat{y}(t+\tau))\hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \Big] \quad (24)\end{aligned}$$

Here, $\nu_f = \frac{1}{\mu_f} = \frac{1-c}{c}$.

3.3. Batch and Semi-Batch

Since we are given $\{x(n)\}_{n=1}^N$ as a set of mixture source vectors, the expectation $E[\cdot]$ is approximated by $\frac{1}{T}\sum_{i=1}^T [\cdot]$ where T is the number of samples in a selected window. The case of $T = N$ is the full batch mode. If we use $T < N$ as a window, it becomes a semi-batch mode. If $T = 1$, the case is an incremental learning. It is possible to choose a window size smaller than N for the look-ahead part so that computation is alleviated. This style of semi-batch mode is recommended for the turbo f-ICA.

4. Alpha-EM Algorithm

The EM algorithm [11] generates a probabilistic model which results from the maximization of the log-likelihood function. In this section, we discuss the use of the α-logarithm (12) as the likelihood ratio to be maximized.

Let $p(x|\psi)$ and $p(x|\varphi)$ be complete-data probability density functions with generic parameters φ and ψ. Let $p(y|\psi)$ and $p(y|\varphi)$ be incomplete-data probability density functions. Denote the α-log likelihood ratio of the incomplete probability densities be $L_Y^{(\alpha)}(\psi|\varphi)$. Define

$$Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) \stackrel{\text{def}}{=} E_{p_{X|Y,\varphi}}\left[L_X^{(\alpha)}(\psi|\varphi) \right]. \quad (25)$$

Let $D^{(\alpha)}(\varphi\|\psi)$ be the α-divergence between $p_{X|Y,\varphi}(x|y,\varphi)$ and $p_{X|Y,\psi}(x|y,\psi)$. Then, we have the following basic equality.

$$\begin{aligned}L_Y^{(\alpha)}(\psi|\varphi) &= Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) \\ &\quad + \tfrac{1-\alpha}{2} \left\{ \frac{p_{Y|\psi}}{p_{Y|\varphi}} \right\}^{\frac{1+\alpha}{2}} D^{(\alpha)}(\varphi\|\psi) \quad (26)\end{aligned}$$

This equality means that the incomplete-data probability density function $p_{Y|\Psi}(y|\psi)$ is maximized in Ψ by transferring this maximization to $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$. Note that the case of $\alpha = -1$ is the traditional EM algorithm.

Starting from Equation (26), the optimization method called the alpha-EM algorithm is given as follows.

[E-step] Compute $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$.

[M-step] Compute $\psi^* = \arg\max_{\psi\in\Psi} p_{Y|\psi}(y|\psi)$. Replace φ by ψ^* and go back to the E-step until the convergence is achieved.

Before closing this section, we comment the following. The alpha-EM algorithm is a soft-decision generalization of the vector quatization [12], [13].

5. Real-World Application of the f-ICA: Brain Activation Map

The purpose of this experiment is to find independent spatial patterns in the brain functional magnetic resonance imaging. Since Equation (16) holds, we can regard each column vector of $U \stackrel{\text{def}}{=} W^{-1}$ be an activation pattern of separated brain maps [14]. The fMRI data are measured by assigning a series of "on-off" stimuli to a tested person. Figure 1 is a resulting brain map which separates the edges of visual regions (V1 and V2). Figure 2 is the corresponding activation pattern. Since usual ICA can not identify the permutation matrix Π, an injection of prior knowledge was incorporated in the update of $W(t+1)$.

6. Concluding Remarks

In this paper, the concept of the optimization transfer was refined and the progress from [2] was explained. Practical applications of the f-ICA were presented on the brain imaging. For the alpha-EM algorithm, simulations can be found in [18].

Acknowledgements

The authors are very grateful to Dr. Keiji Tanaka and Dr. R. Allen Waggoner of RIKEN BRI for permitting them to try out the test data set.

References

[1] I. Csiszár: "Information-type measures of difference of probability distributions and indirect observations," *Studia Sci. Math. Hungarica*, vol. 2, pp. 299-318, 1967.

[2] Y. Matsuyama, S. Imahara and N. Katsumata: "Optimization transfer for computational learning," *Proc. Int. Joint Conf. on Neural Networks*, vol. 3, pp. 1883-1888, 2002.

[3] J. Havrda and F. Charvát: "Qualification method of classification processes: Concept of structural α-entropy," *Kybernetika*, vol. 3, pp. 30-35, 1967.

[4] S. Amari,: "Differential geometry of statistics," Institute of Mathematical Statistics Lecture Notes, vol. 10, pp. 21-94, 1985.

[5] S. Amari and H. Nagaoka: *Methods of Information Geometry*, Iwanami, 1993 (Translation by D. Harada, AMS, 2000).

[6] C. Jutten and J. Herault: "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1-20, 1991.

[7] A.J. Bell and T.J. Sejnowski: "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129-1159, 1995.

[8] H.H. Yang and S. Amari: "Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information," *Neural Computation*, vol. 9, pp. 1457-1482, 1997.

[9] J.-F. Cardoso and B.H. Laheld: "Equivariant adaptive source separation," *IEEE Trans. on SP*, vol. 44, pp. 3017-3030, 1996.

[10] S. Amari, " Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 252-276, 1998.

[11] A.P. Dempster, N.M. Laird and D.B. Rubin: "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. R. Stat. Soc. B*, vol. 39, pp.1-38, 1977.

[12] Y. Matsuyama: "Multiple descent cost competition: Restorable self-organization and multimedia image processing," *IEEE Trans. on Neural Networks*, vol. 9, pp. 106-122, 1998.

[13] Y. Matsuyama, N. Takeda, S. Furukawa and T. Niimoto: "A hierarchy from α-EM algorithm to vector quantization and self-organization," *Proc. Int. Conf. on Neural Information Processing*, vol. 1,pp. 233-238, 1998.

[14] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski: "Spatially independent activity patterns in functional MRI data during the stroop color-naming task, " *Proc. National Academy of Sci. USA*, vol. 95, pp. 803-810, 1998.

[15] Y. Matsuyama and S. Imahara: "The α-ICA algorithm and brain map distillation from fMRI images," *Proc. Int. Conf. on Neural Information Processing*, vol. 2, pp. 708-713, 2000.

[16] Y. Matsuyama, N. Katsumata and S. Imahara: "Independent component analysis using convex divergence," *Proc. Int. Conf. on Neural Information Processing*, vol. 3, pp. 1173-1178, 2001.

[17] Y. Matsuyama and S. Imahara: "Independent component analysis by convex divergence minimization: Applications to brain fMRI analysis," *Proc. Int. Joint Conf. on Neural Networks*, vol. 1, pp. 412-417, 2001.

[18] Y. Matsuyama: "The α-EM algorithm and its basic properties," *Trans. Inst. Electro., Info. and Comm. Engr.*, vol. J82-D-I, pp. 1347-1358, 1999.

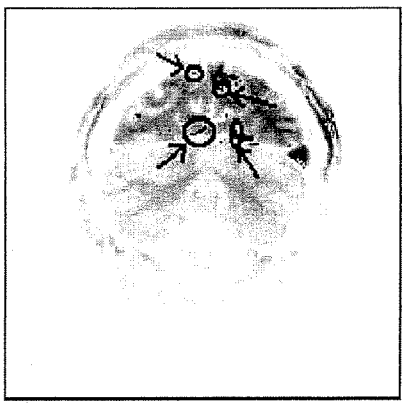


Figure 1: Separation of V1 and V2. Background intensity is inverted.

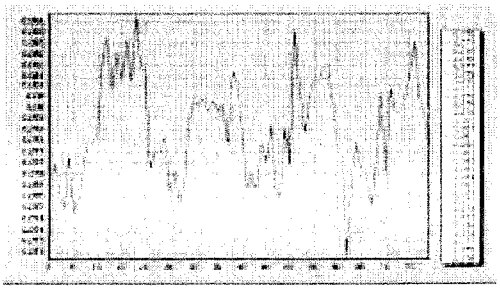


Figure 2: Corresponding activation.

ICONIP'02 SEAL'02 FSKD'02

9th International Conference on Neural Information Processing

4th Asia-Pacific Conference on Simulated Evolution And Learning

1st International Conference on Fuzzy Systems and Knowledge Discovery

November 18-22, 2002, Orchid Country Club, Singapore

ICONIP 2002

SEAL 2002

FSKD 2002

NOTE: The HTML navigation system is provided only for cases where PDF based navigation is not possible. This HTML navigation is lacking many of the functionalities available in PDF. Please use PDF whenever possible.

SUPERVIZED MAP ICA:
APPLICATIONS TO BRAIN FUNCTIONAL MRI

Yasuo Matsuyama and Ryo Kawamura

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan.
yasuo2@waseda.jp, ryo@wizard.elec.waseda.ac.jp

ABSTRACT

This paper gives a method to control or organize itself an activation pattern of fMRI maps obtained by ICA (independent component analysis). The presented method uses an additional term to the convex divergence's gradient. The following merits are observed: (i) Prior knowledge can be effectively used so that obtained activation patterns properly reflect the task on the subject. (ii) Difficulty of finding the appropriate activation pattern due to the permutation can be avoided. Experiments on brain fMRI maps for visual cortices are tried and reported.

1. INTRODUCTION

Independent component analysis (ICA) has found fruitful applications in composite signal separation [1], [2] including brain functional MRI [3].

There are two types of iterative algorithms for ICA. One is a method to add a small descent cost vector [1], [2], [4]. The other is a fixed point method [5]. The former, the descent cost method, is amenable to incorporate additional adaptive terms. Besides, the faster method for this class is presented to improve the convergence speed; the f-ICA method [6], [7].

The ICA is based on optimization of a cost function which reflects independence of separated components. But, most of optimization methods are often trapped by local optima. Even if we could get appropriate independent components, the permutation, i.e., their ordering remains indefinite. Thus, a special care is necessary to judge if the generated patterns are proper. Because of this property, we need a method to find task-related and well-ordered "independent patterns. Thus, the method presented in this paper is a class of prior knowledge injection [8], [9], [10]. The main feature of this paper's method is as follows.

(i) The form of the prior knowledge is expressed simply.

(ii) The update method matches to gradient style learning.

(iii) Need of matrix inversion is restricted [9].

The organization of this paper is as follows. Section 2 gives a formulation of the independent component analysis and the derivation of the f-ICA algorithm. Section 3 explains a method to incorporate the prior knowledge as supervisory information into the ICA's update term. The method is a generalization and refinement of the fMRI map distillation given in [9]. Section 4 shows effects of the regularization term for the supervisory information through experiments on visual cortices. Effectiveness of the presented method is reported. Section 5 is provided for concluding remarks.

2. THE f-ICA: A FASTER GENERALIZATION OF THE MINIMUM MUTUAL INFORMATION ICA

2.1. Formulation of the ICA

In the problem of ICA, we are given a set of vector random variables.

$$x(n) = \mathrm{col}[x_1(n), \cdots, x_K(n)] = As(n), \quad (n = 1, \cdots, N). \tag{1}$$

Here, the matrix A and the vector

$$s(n) = \mathrm{col}[s_1(n), \cdots, s_K(n)] \tag{2}$$

are unknown except that

1. The components $s_i(n)$ and $s_j(n)$ are independent each other for $i \neq j$.

2. The components $s_i(n)$, $(i = 1, \cdots, K)$, are non-Gaussian except for at most one i.

Vol. 5 , pp. 2259-2263 , 2002

Under such conditions, the learning algorithm for the ICA finds a demixing matrix

$$W = \Lambda \Pi A^{-1} \tag{3}$$

so that the components of

$$Wx(n) \stackrel{\text{def}}{=} y(n) = \text{col}[y_1(n), \cdots, y_K(n)] \tag{4}$$

are independent each other for every n. Here,

$$\Lambda = \text{diag}[d_1, \cdots, d_K] \tag{5}$$

is a nonsingular diagonal matrix, and Π is a permutation matrix. Both of them are also unknown. We will estimate the demixing matrix W by a gradient method for minimizing a cost function on the degree of independence.

2.2. Update Term Derived from the Convex Divergence

Let $p(y) = p(y_1, \cdots, y_K)$ be a joint probability density, and $q(y) = \prod_{i=1}^K q_i(y_i)$ be a product probability density. Then, the independence is obtained by the minimization of the following cost.

$$\begin{aligned} I_f(\bigwedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} D_f\left(p(y_1, \cdots, y_K) \| \prod_{i=1}^K q_i(y_i)\right) \\ &\stackrel{\text{def}}{=} D_f(p(y)\|q(y)) \\ &= D_g(q(y)\|p(y)) \\ &= I_g(\bigwedge_{i=1}^K Y_i) \\ &= \int_{\mathcal{X}} p(x) g\left(\frac{|W|q(y)}{p(x)}\right) dx. \end{aligned} \tag{6}$$

Here, $D_f(p\|q)$ is the convex divergence [11] between p and q in terms of the twice differentiable convex function $f(r)$ with $f(1) = 0$. The dual function g is defined by

$$g(r) = rf(1/r). \tag{7}$$

The symbol "$\bigwedge$" is used instead of ";" adopted in information theory textbooks. Then, we have

$$\begin{aligned} &-\tilde{\nabla} I_g(\bigwedge_{i=1}^K Y_i) \stackrel{\text{def}}{=} -\frac{\partial I_g(\bigwedge_{i=1}^K Y_i)}{\partial W}(cW^TW) \\ &= -c \int_{\mathcal{X}} q(y) g'\left(\frac{|W|q(y)}{p(x)}\right) \{I - \varphi(y) x^T W^T\} |W| dx\, W \\ &= -c \int_{\mathcal{Y}} q(y) g'\left(\frac{q(y)}{p(y)}\right) \{I - \varphi(y) y^T\} dy\, W. \end{aligned} \tag{8}$$

Here,

$$\varphi(y) = -\text{col}\left[\frac{q_1'(y_1)}{q_1(y_1)}, \ldots, \frac{q_K'(y_K)}{q_K(y_K)}\right], \tag{9}$$

and

$$c \stackrel{\text{def}}{=} \frac{f''(1)}{f'(1)} = -\frac{g''(1)}{g'(1)} \in (-\infty, \infty). \tag{10}$$

Since

$$qg'(q/p) = -g''(1)p + \{g'(1) + g''(1)\}q + o(1) \tag{11}$$

around $p \approx q$, we have the following equations:

$$\begin{aligned} &-\frac{\partial I_f}{\partial W}(cW^TW) = -\frac{\partial I_g}{\partial W}(cW^TW) \\ &= f''(1)\Big[c\left\{I - E_{p(y)}[\varphi(y)y^T]\right\} W \\ &\quad + (1-c)\left\{I - E_{q(y)}[\varphi(y)y^T]\right\} W\Big] + o(1). \end{aligned} \tag{12}$$

In our case, the natural gradient [12] is used in the form as the above by multiplying cW^TW. Then, the update term is

$$\tilde{\Delta}_f W = -\rho(t) \frac{\partial I_f}{\partial W} W^T W. \tag{13}$$

Here, $\rho(t)$ is a small positive number called the learning rate.

2.3. Implementation of the f-ICA

2.3.1. Momentum f-ICA

Since $p(y)$ and its target $q(y)$ appears in equation (12), software implementation requires interpretations of $p(y)$ and $q(y)$. First, we consider a non-anticipatory approximation at the t-th iteration such that

$$p(y) \Leftarrow p^{(t-\tau)}(y) \quad \text{and} \quad q(y) \Leftarrow p^{(t)}(y). \tag{14}$$

By this method, we have the following sample-based learning algorithm.

[Momentum f-ICA]

If we use $p(y)$ as $p^{(t-\tau)}(y)$ and $q(y)$ as $p^{(t)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned} &\tilde{\Delta}_f W(t) = \tilde{\Delta} W(t) + \mu_f \tilde{\Delta} W(t-\tau) \\ &= \rho_t\Big[\{I - \varphi(y(t))y(t)^T\} W(t) \\ &\quad + \mu_f \{I - \varphi(y(t-\tau))y(t-\tau)^T\} W(t-\tau)\Big] \end{aligned} \tag{15}$$

Here, $\mu_f = \frac{c}{1-c}$.

2.3.2. Look-ahead f-ICA

There is an anticipatory approximation at the t-th iteration such that

$$p(y) \Leftarrow p^{(t)}(y) \quad \text{and} \quad q(y) \Leftarrow p^{(t+\tau)}(y). \tag{16}$$

This is conceptually more natural than the momentum f-ICA since $p(y)$ has the present iteration index. Then, we have the following sample-based learning algorithm.

[Look-ahead (Turbo) f-ICA]

$$\tilde{\Delta}_f W(t) = \tilde{\Delta} W(t) + \nu_f \tilde{\Delta} W(t+\tau)$$
$$= \rho_t \Big[\{I - \varphi(y(t)) y(t)^T\} W(t)$$
$$+ \nu_f \{I - \varphi(\hat{y}(t+\tau)) \hat{y}(t+\tau)^T\} \hat{W}(t+\tau) \Big] \quad (17)$$

Here, $\nu_f = \frac{1}{\mu_f} = \frac{1-c}{c}$.

We comment here that $\tau = 1$ works effectively enough for both anticipatory and non-anticipatory methods. It is worthy to note here that a semi-batch mode with a window size less than N is recommended for the look-ahead ICA so that computational load is alleviated.

3. CONTROL OF THE DEMIXING MATRIX BY SUPERVISORY INFORMATION

3.1. ICA Maps and Activation Patterns

From equation (4), we have

$$x(n) = W^{-1} y(n) \stackrel{\text{def}}{=} U y(n). \quad (18)$$

Here,

$$U \stackrel{\text{def}}{=} [u_1, \cdots, u_K], \quad (19)$$

where each u_k is a column vector. From Equations (18) and (19), we can understand the following.

(i) The observed signal $x(n)$ is generated by the mixing matrix U from the estimated signal $y(n)$.

(ii) Each column vector

$$u_k = \text{col}[u_{1k}, \cdots, u_{Kk}] \quad (20)$$

stands for the mixing weight pattern for the unknown independent components.

Note that from Equations (1), (3) and (4), we have

$$\Lambda \Pi s(t) = y(t). \quad (21)$$

This leads to the following idea.

(i) The permutation matrix is implicitly incorporated by selecting an appropriate column index k so that u_k is adjusted by supervisory information.

(ii) Following the selection of the index k, the signal level d_k of the supervisory pattern is adjusted to u_k.

Thus, supervisory information reflecting the designated task can be incorporated into the f-ICA through organization of the vector u_k.

3.2. Injection of Supervisory Information

The supervisory information is injected by

$$U(t+1) = U(t) + \Delta U(t). \quad (22)$$

We derive the increment $\Delta U(t)$ from the minimization of

$$F(U, \hat{R}) = tr\{(\hat{R} - U)^T (\hat{R} - U)\}. \quad (23)$$

Here, $\hat{R}$ is the target pattern decided from the experimental task applied to the subject. The cost function (23) watches if U is close to the target pattern $\hat{R}$. "Trace" appears since U and $\hat{R}$ are matrices. Then,

$$\Delta U = -\frac{1}{2} \lambda \frac{dF(U, \hat{R})}{dU}. \quad (24)$$

Since the main part of the ICA's increment is Equation (13), we need the following $\Delta V(t)$ which corresponds to $\Delta U(t)$:

$$U^{-1}(t+1) = U^{-1}(t) + \Delta V(t), \quad (25)$$

i.e.,

$$W(t+1) = W(t) + \Delta V(t). \quad (26)$$

For this computation, we use the following formula: For $B = A + Y$,

$$B^{-1} = A^{-1} - A^{-1}(I + YA^{-1})^{-1} YA^{-1} \quad (27)$$

holds. Then, we have

$$W(t+1) = W(t) - W(t)\{I + \Delta U(t) W(t)\}^{-1} \Delta U(t) W(t). \quad (28)$$

Note that

$$\{I + \Delta U(t) W(t)\}^{-1} \approx -\Delta U(t) W(t) \quad (29)$$

if $\|\Delta U(t) W(t)\| << 1$. Then, the update for minimizing $F(U, \hat{R})$ is

$$W(t+1) = W(t) - W(t) \Delta U(t) W(t). \quad (30)$$

That is, the additional term to Equation (13) is

$$\Delta V(t) = -W(t) \Delta U(t) W(t). \quad (31)$$

Then, the information of the teacher pattern $\hat{R}$ is suitably reflected in the ICA learning algorithm.

Computation of the increment (24) can be made exactly or approximately.

(i) From Equations (23) and (24), one obtains

$$\Delta U = \lambda(\hat{R} - U) = \lambda\hat{R}\{I - (W\hat{R})^{-1}\}. \quad (32)$$

(ii) Further computation of Equation (32) using

$$I - (W\hat{R})^{-1} \approx W\hat{R} - I \quad (33)$$

for

$$\|I - (W\hat{R})^{-1}\| << 1, \quad (34)$$

gives

$$\Delta U = \lambda\hat{R}(W\hat{R} - I). \quad (35)$$

Here, the higher order term is omitted.

3.3. Teacher Pattern Assignment

As was explained in the previous section, the supervisory information is injected to the matrix U by specifying the task pattern $\hat{R}$. This supervision is column-wise. Let a column vector

$$\hat{a}_k = \text{col}[0,0,0,1,1,1,0,0,0,\cdots,1,1,1,0,0,0], \quad k \in \{0,\cdots,K-1\} \quad (36)$$

be an on-off pattern of the assigned task to the subject. Then, we compute its power-matched version $\hat{r}_k$ where the column sum is zero and the variance is the same as u_k. Then, Δu_k is computed by using Equation (32) or (35). If the rest $\{\hat{r}_j\}$, $j \neq k$, is arbitrary, i.e., unsupervised, this freedom is interpreted as $\Delta u_j = 0$ for $j \neq k$. Note that

(i) Selecting k is a process of finding an appropriate permutation.

(ii) The power matching reduces the amplitude's uncertainty d_k in Equation (5).

4. ACTIVATION PATTERNS OF BRAIN fMRI MAPS

4.1. Effect of the Teacher Pattern and Learning Rate Control

First, we check to see how the usage of the teacher pattern is effective. Figure 1 is the resulting activation pattern when the learning rate $\lambda = \lambda(t)$ in Equation (32) is kept constant. As is illustrated in this figure, obtained activation pattern strongly reflects the teacher signal of the on-off time course. Thus, the supervisory information is effectively injected to such time course organization. But, we found the following: Since we want to find a set of independent patterns, a learning rate $\lambda(t)$ kept to be a large constant may contradict to the components' independence. Therefore,

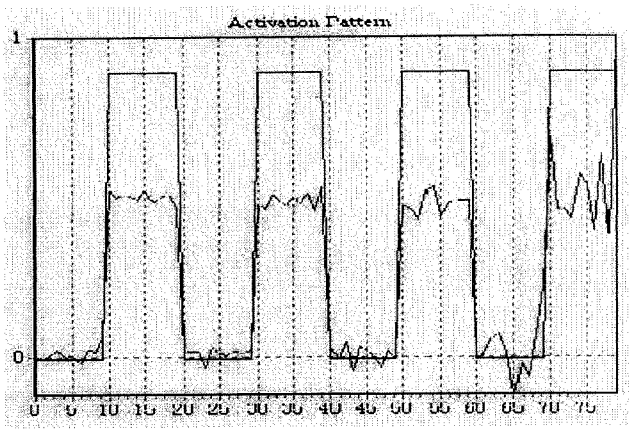


Figure 1: A time course with $\lambda(t) = const.$

we select a method that $\lambda(t)$ is gradually reduced to zero or to its vicinity.

4.2. Distillation of Task-Related Maps

The experimental task is as follows. Three visual patterns are shown to a subject. They are as follows.

(r) A dark background with a small red cross at the center.

(s) A still image with many white squares located randomly.

(m) Two groups of moving squares in opposite directions each other.

Our goal is to distill the following.

(i) A meaningful activation pattern which is relevant to moving image recognition.

(ii) Its associated brain map.

Figure 2 is the resulting activation pattern obtained by this paper's supervised ICA (actually, supervised f-ICA). We can see that this pattern reflects switching in the task clearly. Figure 3 illustrates the corresponding map to the activation pattern of Figure 2. The encircled region is identified to be the most active area located in the *dorsal occipital cortex* in the right hemisphere. This region appears much more clearly than the previous study [9] because of the effectiveness of the supervisory ICA (f-ICA).

5. Concluding Remarks

In this paper, we presented a method to introduce a strategy of supervised learning to ICA. The supervisory concept was injected to ICA through its activation pattern. This strategy matches well to the f-ICA, i.e., to the faster version of the gradient update

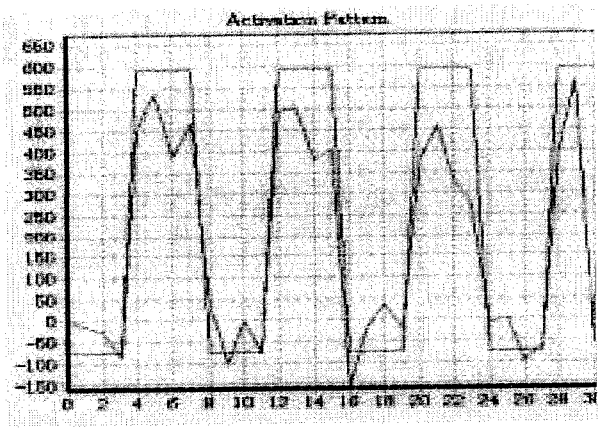


Figure 2: Task-related activation pattern.

method. The presented method is related to regul: ization and self-organization.

In general. it is highly likely in real data that class of hints is accompanied. In such a case, this information can be incorporated into ICA so that better and compatible results are obtained. Thus, this paper presented guidelines to incorporate such hints as the supervisory information for the ICA.

ACKNOWLEDGMENTS

The authors are grateful to Dr. R. Allen Waggoner and Dr. Keiji Tanaka of RIKEN BRI for permitting them to try out the test data set. Mr. Suichiro Imahara of Toshiba receives thanks for helping the authors in initial study of this paper. This work was supported by High-Technology/New-Technology Research Projects of Waseda University, and Grants-in-Aids for Scientific Research #13680465.

References

[1] C. Jutten and J. Herault, Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing, vol. 24, pp. 1-20, 1991.

[2] A.J. Bell and T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[3] M.J. McKeown, T-P. Jung, S. Makeig, G. Brown, S.S. Kindermann, T-W. Lee and T.J. Sejnowski, Spatially independent activity patterns in functional MRI data during the Stroop color-naming task, Proc. Natl. Acad. Sci. USA, vol. 95, pp. 803-810, 1998.

[4] H.H. Yang and S. Amari, Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

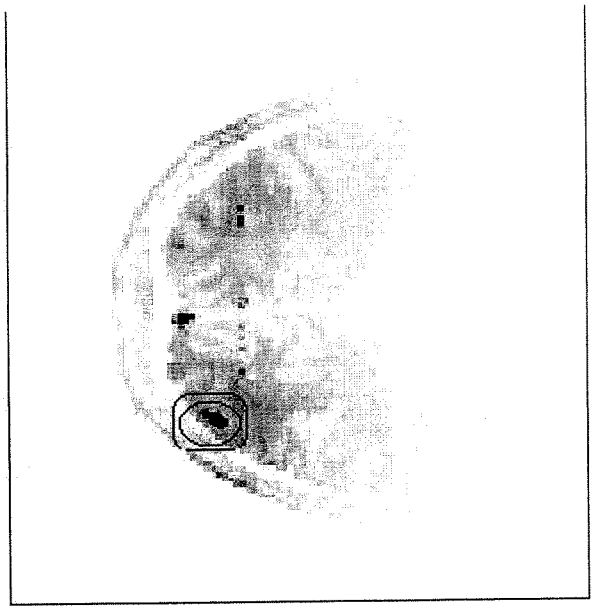


Figure 3: Corresponding brain map which identifies an area responding to the moving image. Background anatomical data's gray level is inverted.

[5] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. on Neural Networks, Vol. 10, pp. 626-634, 1999.

[6] Y. Matuyama, N. Katsumata, Y. Suzuki and S. Imahara, The α-ICA algorithm, Proc. ICA2000, pp. 297-302, 2000.

[7] Y. Matsuyama, N. Katsumata and S. Imahara, Independent component analysis using convex divergence, Proc. ICONIP2001, pp. 1173-1178, 2001.

[8] A. Hyvärinen and Raju Karthikesh, Sparse priors on the mixing matrix in independent component Analysis, Proc. ICA2000, pp. 477-482, 2000.

[9] Y. Matsuyama and S. Imahara, The alpha-ICA algorithm and brain map distillation from fMRI images, Proc. ICONIP, pp. 708-713, 2000.

[10] W. Lu and J.C. Rajapakse, ICA with reference, Proc. ICA2001, pp. 120-125, 2001.

[11] I. Csiszár, Information-type measures of difference of probability distributions and indirect observations, Studia Sci. Math. Hungarica, vol. 2, pp. 299-318, 1967.

[12] S. Amari, Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

China – Japan – Korea Joint Workshop on Neurobiology and Neuroinformatics (NBNI-2001)

Zhejiang Hotel, Hangzhou, China

November 20-22, 2001

Co-sponsored by:
School of Life Sciences (SLS) & Brain Science Research Center (BSRC), Fudan University (FDU), China
Institute of Neuroscience (ION), Chinese Academy of Sciences (CAS), China
Zhejiang University (ZJU), China
Brain Science Institute (BSI) at RIKEN, Japan
Brain Science Research Center (BSRC) at KAIST, Korea
Biomedical Brain Research Center (BBRC) at KNIH, Korea.

Organizers:
Shun-ichi Amari (RIKEN BSI, Japan)
Chang-Rak Choi (KNIH BBRC, Korea)
Fanji Gu (FDU, China)
Aike Guo (CAS ION, China)
Soo-Young Lee (KAST BSRC, Korea)
Nobuyuki Nukina (RIKEN BSI, Japan)
T. Oh (KIST, Korea)
Qingye Tong (ZJU, China).

Supported by:
Science & Technology Committee of Shanghai Municipality
National Natural Science Foundation of China (NSFC)
Neuroinformatics Center, PLA General Hospital, China

Index

Optimization Transfer in Learning Algorithms: With examples on pattern extraction and emotional coding

Yasuo MATSUYAMA
Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan

Most learning algorithms handle statistical or probabilistic measures of actual and tentative data. For continuous data, the probability measures are in the form of probability density functions. Such probability density functions have generic parameters which are closely related to source patterns. But, such parameters are not explicitly given in practical situations. Only a partial set of raw data is given. Therefore, we have to estimate the generic parameters based upon the observed data. Such a problem is often coined into optimization problems towards learning algorithms. In this situation, logarithm of the probability density is mostly used. This is because the logarithm reflects statistical and information theoretic principles, and alleviates computation. But, the original target for the optimization is the probability density functions per se. Thus, the use of the logarithm already transfers the original optimization target. In this talk and memo, discussed first is the usage of a class of optimization transfer functions which contains the logarithm as a special case. This class of functions comes from discussions on the convex divergence which measures similarity of two probability densities. From this start point, the talk proceeds as follows: (i) Convex divergence, (ii) Optimization transfer, (iii) Fisher measure of information, (iv) Independent component analysis and extraction of brain activity, (v) Expectation-maximization, (vi) Special cases of the expectation-maximization with examples on self-organization and emotional coding

The α-EM Algorithm: Surrogate Likelihood Maximization Using α-Logarithmic Information Measures

Yasuo Matsuyama, *Fellow*

Abstract— **A new likelihood maximization algorithm called the α-EM algorithm (α-Expectation-Maximization algorithm) is presented. This algorithm outperforms the traditional or logarithmic EM algorithm in terms of convergence speed for an appropriate range of the design parameter α. The log-EM algorithm is a special case corresponding to $\alpha = -1$. The main idea behind the α-EM algorithm is to search for an effective surrogate function or a minorizer for the maximization of the observed data's likelihood ratio. The surrogate function adopted in this paper is based upon the α-logarithm which is related to the convex divergence. The convergence speed of the α-EM algorithm is theoretically analyzed through α-dependent update matrices and illustrated by numerical simulations. Finally, general guidelines for using the α-logarithmic methods are given. The choice of alternative surrogate functions is also discussed.**

Index Terms— **α-EM algorithm, α-logarithm, convex divergence, convergence speed, surrogate function, minorization-maximization, exponential family, supervised and unsupervised learning, vector quantization, independent component analysis.**

I. Introduction

THE Expectation-Maximization (EM) algorithm [1], [2], [3], [4], [5] is a popular tool for iteratively maximizing likelihood functions. The designer of an EM algorithm chooses a hypothetical "complete" data set to complement the observed "incomplete" data set. One tries to find an augmented complete data set which, if available, would make the estimation problem easier. The incomplete-data log-likelihood is iteratively increased by maximizing the conditional expectation of the complete-data log-likelihood given the incomplete data and the parameter estimates from the previous iteration. This process can be understood as maximization transfer, surrogate maximization or minorization-maximization [6], [7] since the incomplete-data log-likelihood is increased through the maximization of the complete-data log-likelihood's conditional expectation.

Manuscript received December 2, 1998; revised January 17 and July 1, 2000; January 22, 2001; and October 1, 2002. This work was supported in part by a Grant-in-Aid for Scientific Research #13680465, the High Technology/New Technology Research Funds of Waseda University, and the Productive ICT Academia Program of the 21st Century COE of Japan. The material in this paper was presented at the International Work-Conference on Artificial and Natural Neural Networks, Lanzarote, Spain, June 4-6, 1997, and at the International Symposium on Information Theory, Cambridge, MA, August 16-21, 1998.

The author is with the Department of Computer Science, Waseda University, Tokyo 169-8555, Japan (e-mail: yasuo2@waseda.jp).

Publisher Item Identifier S xxxx-xxxx(xx)xxxxx-x.

The use of the logarithm usually alleviates the complexity of mathematical expressions; however, the convergence speed of the EM algorithm is often slow. Thus, we address the following problem: Is there any class of convexity controllable functions which outperform the log-likelihood optimization? As a trade-off, mathematical structures can be a little more complex than in the logarithmic case.

Logarithms have important roles besides simplifying the likelihood maximization. Information-theoretic derivations of entropy, Kullback-Leibler divergence and the Fisher information matrix all bring about the logarithm [8]. Among these information measures, the Kullback-Leibler divergence was a key for realizing the maximization transfer in the EM algorithm [1]: This motivates us to explore more effective surrogate functions. Our approach is based on an extension of the logarithm associated with a class of general divergence measures [9], [10], [11], [12], [13], [14], [15] which includes the logarithmic or Kullback-Leibler divergence as a specific case.

The organization of the main text is as follows. In Section II, a general class of functions termed α-logarithms is derived via Csiszár's convex divergence [11]. Then, extensions of Fisher scores and information matrices are presented. In Section III, the α-logarithm is used to define α-log-likelihood ratios and the extended EM algorithm, called the α-EM algorithm or WEM (α-Weighted EM algorithm). Then, the α-EM algorithm is extended in parallel to the original log-EM algorithm. Convergence conditions are then discussed. In Section IV, the speed of the α-EM algorithm is analyzed using update matrices. Therein, the speedup is understood by α-dependent changes of spectral radii. In Section V, computing methods on Gaussian mixtures are presented. Examples show that the α-EM family outperforms logarithmic methods in speed as measured by iteration counts and CPU time. Section VI gives concluding remarks and discusses other possible surrogate functions.

II. Convex Divergence, α-Logarithm and Related Information Measures

A. Convex Divergence and α-Logarithm

Consider the following class of divergence measures presented by Csiszár [11][1]. Let $\mathcal{X}$ be an Euclidean space with

[1] An equivalent form is also given in Rényi's work; Equation (4.20) of [9].

arbitrary dimension. Let μ be a measure on a Borel measurable space $(\mathcal{X}, \mathcal{B})$. Let $f : \boldsymbol{R}^+ \to \boldsymbol{R}$ be a convex function, and $p(x)$ and $q(x)$, $x \in \mathcal{X}$, be probability densities with respect to μ. Then,

$$D_C(p\|q) \stackrel{\text{def}}{=} \int_{\mathcal{X}} q\, f(p/q) d\mu(x) \geq f(1). \tag{1}$$

This quantity is called f-divergence [11], φ-divergence [15], or convex divergence. We assume that f is strictly convex at $r = 1$ and that $f(1) = 0$. This assures that $D_C(p\|q) \geq 0$ with equality if and only if $p(x) = q(x)$ for μ-almost all x. Unless there is a possibility of confusion, the symbol $d\mu(x)$ is simply denoted by dx throughout the text.

There is a variety of such convex functions of interest [11], [12], [13], [14], [15]. In this paper, we consider the following convex function for $\alpha \neq \pm 1$:

$$\begin{aligned} f^{(\alpha)}(r) &= \tfrac{4}{1-\alpha^2}(r - r^{\frac{1-\alpha}{2}}) \\ &= \left\{\tfrac{2}{1-\alpha} r^{\frac{1-\alpha}{2}}\right\} \left\{\tfrac{2}{1+\alpha}(r^{\frac{1+\alpha}{2}} - 1)\right\} \end{aligned} \tag{2}$$

with $r = p/q$. Since p and q are probability densities, $D_C(p\|q)$ is expressed as

$$\begin{aligned} D^{(\alpha)}(p\|q) &= \tfrac{4}{1-\alpha^2}\left\{1 - \int_{\mathcal{X}} q(p/q)^{\frac{1-\alpha}{2}} dx\right\} \\ &= \tfrac{4}{1-\alpha^2}\left\{1 - \int_{\mathcal{X}} p(q/p)^{\frac{1+\alpha}{2}} dx\right\}, \end{aligned} \tag{3}$$

henceforth referred to as α-divergence. We have the skew symmetry

$$D^{(\alpha)}(p\|q) = D^{(-\alpha)}(q\|p). \tag{4}$$

In the limiting case of $\alpha = \pm 1$, we have $f^{(-1)}(r) = r \log r$ and

$$D^{(-1)}(p\|q) = \int_{\mathcal{X}} p \log(p/q)\, dx = D^{(1)}(q\|p). \tag{5}$$

Here, the base of the logarithm is 'e'. The quantity (5) is the Kullback-Leibler divergence[2] which plays a central role in information theory [8] and in the derivation of the EM-algorithm [1]. The α-divergence is an information measure and may be derived from a set of axioms [9][3], [10]. Our α-EM algorithm will require $\alpha \in (-\infty, 1)$.

Due to the factorization (2) of the convex function $f^{(\alpha)}(r)$, consider

$$L^{(\alpha)}(r) = \tfrac{2}{1+\alpha}(r^{\frac{1+\alpha}{2}} - 1) \tag{6}$$

[16], [17], [18], [19], [20], which we call the α-logarithm[4].

[2]Well-known special cases other than $\alpha = \pm 1$ are the Hellinger distance, $D^{(0)}(p\|q) = 2\int_{\mathcal{X}}(\sqrt{p} - \sqrt{q})^2 dx = 2\|\sqrt{p} - \sqrt{q}\|^2$, and the weighted square distance, $D^{(-3)}(p\|q) = \frac{1}{2}\int_{\mathcal{X}}(p-q)^2/p\, dx$.

[3]Rényi's α-divergence is

$$\begin{aligned} D_R^{(\alpha)}(p\|q) &\stackrel{\text{def}}{=} -\tfrac{4}{1-\alpha^2}\log\{\int_{\mathcal{X}} q(p/q)^{\frac{1-\alpha}{2}} dx\} \\ &= -\tfrac{4}{1-\alpha^2}\log\{1 - \tfrac{1-\alpha^2}{4} D^{(\alpha)}(p\|q)\}. \end{aligned}$$

Therefore, $D_R^{(\alpha)}(p\|q)$ is equivalent to $D^{(\alpha)}(p\|q)$ in the sense of optimization due to the monotone increasing property of the logarithm.

[4]As an alternative to (2), the α-logarithm can be obtained form its dual convex function:

$$g^{(\alpha)}(r) \stackrel{\text{def}}{=} r f^{(\alpha)}(1/r) = \tfrac{-2}{1-\alpha} L^{(\alpha)}(r).$$

Lemma 1: For $\alpha \in (-\infty, \infty)$ and $r \in (0, \infty)$, the α-logarithm $L^{(\alpha)}(r)$ satisfies the following properties.

(i) $L^{(-1)}(r) = \log r \stackrel{\text{def}}{=} \ell(r)$.

(ii) $L^{(\alpha)}(1) = 0$.

(iii) $L^{(\alpha)}(r)$ is strictly monotone increasing with respect to r.

(iv) $L^{(\alpha)}(r)$ is strictly concave for $\alpha < 1$; a straight line $r-1$ for $\alpha = 1$; and strictly convex for $\alpha > 1$.

(v) $L^{(\alpha)}(r) \leq L^{(\beta)}(r)$ for $\alpha < \beta$, where equality holds if and only if $r = 1$.

(vi) (α-log-sum inequalities)
Let $\{a_i\}_{i=1}^n$ and $\{b_i\}_{i=1}^n$ all be non-negative numbers. Then, the following three cases hold.
If $\alpha > -3$:

$$\begin{aligned} &\textstyle\sum_{i=1}^n a_i L^{(\alpha)}(a_i/b_i) \\ &\qquad \geq (\textstyle\sum_{i=1}^n a_i) L^{(\alpha)}(\sum_{i=1}^n a_i / \sum_{i=1}^n b_i). \end{aligned}$$

The equality holds if and only if $a_i/b_i = constant$. The case $\alpha = -1$ is the log-sum inequality [8].
If $\alpha = -3$: The inequality "$\geq$" is replaced by the equality "$=$".
If $\alpha < -3$: The inequality "$\geq$" is replaced by the reverse inequality "$\leq$", whose equality holds if and only if $a_i/b_i = constant$.

(vii) (chain inequality)
Let a, b, c and d all be positive. Let $L^{(\alpha)}(b/a) \geq 0$, $L^{(\alpha)}(c/b) \geq 0$ and $L^{(\alpha)}(d/c) \geq 0$. Then,

$$0 \leq L^{(\alpha)}(b/a) \leq L^{(\alpha)}(c/a) \leq L^{(\alpha)}(d/a) \tag{7}$$

holds. The right equality of (7) holds if and only if $d = c$.

(viii) Let ψ be a finite dimensional vector in an Euclidean space. If $r = r(\psi) \in (0, \infty)$ is twice differentiable with respect to ψ, the following equalities hold:

$$\frac{\partial L^{(\alpha)}(r)}{\partial \psi} = r^{\frac{1+\alpha}{2}} \frac{\partial \log r}{\partial \psi}, \tag{8}$$

$$-\frac{\partial^2 L^{(\alpha)}(r)}{\partial \psi \partial \psi^T} = r^{\frac{1+\alpha}{2}} \left\{\left(-\frac{\partial^2 \log r}{\partial \psi \partial \psi^T}\right) - \frac{1+\alpha}{2} \frac{\partial \log r}{\partial \psi} \frac{\partial \log r}{\partial \psi^T}\right\}. \tag{9}$$

Note that $\frac{\partial}{\partial \psi}$ is used as an equivalent notation for the gradient ∇_ψ in this paper.

The α-log-sum inequalities are obtained similarly to [8]. Other items come directly from (6). Note that, as in (iv), the range $\alpha < 1$ guarantees concavity of $L^{(\alpha)}(r)$.

The inverse of the α-logarithm is the α-exponential

$$E^{(\alpha)}(r) \stackrel{\text{def}}{=} (1 + \tfrac{1+\alpha}{2} r)^{\frac{2}{1+\alpha}}$$

which satisfies

$$L^{(\alpha)}(E^{(\alpha)}(r)) = E^{(\alpha)}(L^{(\alpha)}(r)) = r.$$

Note that $U^{(\alpha)}(r) = \frac{2}{1-\alpha} r^{\frac{1-\alpha}{2}}$ includes r as the special case of $\alpha = -1$. The inverse of this α-linear function is $V^{(\alpha)}(r) = (\frac{1-\alpha}{2} r)^{\frac{2}{1-\alpha}}$ which satisfies

$$U^{(\alpha)}(V^{(\alpha)}(r)) = V^{(\alpha)}(U^{(\alpha)}(r)) = r.$$

B. α-Versions of Important Statistical Measures

Let $p_{X|\Psi}(x|\psi)$ be a probability density of a vector random variable X, parameterized by a finite-dimensional vector $\psi \in \mathcal{R} \subset \boldsymbol{R}^d$. We consider a family of probability densities $\mathcal{F}_{\mathcal{R}} \stackrel{\text{def}}{=} \{p_{X|\Psi}(x|\psi)\}_{\psi\in\mathcal{R}}$ satisfying the following regularity conditions [12], [21], [22] [5]:

(a) $p_{X|\Psi}(x|\psi) > 0$ for μ-almost all x and all $\psi \in \mathcal{R}$.
(b) $p_{X|\Psi}(x|\psi)$ is twice differentiable with respect to each coordinate of ψ, for μ-almost all x.
(c) $\int_{\mathcal{X}} p_{X|\Psi}(x|\psi)d\mu(x)$ exists and can be differentiated up to twice under the integral sign with respect to each coordinate of ψ [6].
(d) The Fisher information matrix is finite and positive definite, i.e.,

$$F_X(\psi) \stackrel{\text{def}}{=} E_p\left[\frac{\partial \ell_X(\psi)}{\partial\psi}\frac{\partial \ell_X(\psi)}{\partial\psi^T}\right] = -E_p\left[\frac{\partial^2 \ell_X(\psi)}{\partial\psi\partial\psi^T}\right] > 0. \tag{10}$$

Here, $\ell_X(\psi)$ stands for $\log p_{X|\Psi}(x|\psi)$, and $E_p[\cdot]$ means $\int_{\mathcal{X}} p_{X|\Psi}(x|\psi)[\cdot]d\mu(x)$.

The regularity conditions (a) $\sim$ (d) on the family $\mathcal{F}_{\mathcal{R}}$ will be required throughout this paper.

The α-log-likelihood for $p_{X|\Psi}(x|\psi)$ is defined by

$$L_X^{(\alpha)}(\psi) \stackrel{\text{def}}{=} L^{(\alpha)}\left(p_{X|\psi}(x|\psi)\right) = \frac{2}{1+\alpha}\{p_{X|\psi}(x|\psi)^{\frac{1+\alpha}{2}} - 1\} \tag{11}$$

and is often denoted by $L^{(\alpha)}$ or L. Also, $p_{X|\psi}(x|\psi)$ will often be denoted simply by $p(x|\psi)$ or p. From (8), the α-efficient score is

$$\begin{aligned}\frac{\partial L_X^{(\alpha)}(\psi)}{\partial\psi} &= p_{X|\psi}(x|\psi)^{\frac{1+\alpha}{2}}\frac{\partial \log p_{X|\psi}(x|\psi)}{\partial\psi} \\ &= p_{X|\psi}(x|\psi)^{\frac{1+\alpha}{2}}\frac{\partial \ell_X(\psi)}{\partial\psi}.\end{aligned} \tag{12}$$

The vector $\partial\ell_X(\psi)/\partial\psi$ is the original Fisher's efficient score.

Next, we consider an information matrix using the α-log-likelihood

$$\begin{aligned}M_X^{(\alpha)}(\psi) &\stackrel{\text{def}}{=} E_p\left[\frac{1-\alpha}{2}p^{-(1+\alpha)}\left(\frac{\partial L^{(\alpha)}}{\partial\psi}\right)\left(\frac{\partial L^{(\alpha)}}{\partial\psi^T}\right)\right] \\ &= -E_p\left[p^{-\frac{1+\alpha}{2}}\left(\frac{\partial^2 L^{(\alpha)}}{\partial\psi\partial\psi^T}\right)\right].\end{aligned} \tag{13}$$

If $\alpha = -1$, this α-information matrix reduces to the usual Fisher information matrix [7], $M_X^{(-1)}(\psi) = F_X(\psi)$. Under the regularity conditions (a) $\sim$ (d), $M_X^{(\alpha)}(\psi)$ is guaranteed to be finite for $-\infty < \alpha < \infty$, and

$$M_X^{(\alpha)}(\psi) = \frac{1-\alpha}{2}F_X(\psi) \stackrel{\text{def}}{=} m(\alpha)F_X(\psi). \tag{14}$$

Thus, the α-information matrix $M_X^{(\alpha)}(\psi)$ is proportional to the usual Fisher information matrix $F_X(\psi)$. We call the coefficient $m(\alpha)$ the scale factor because of the relationship to the derivatives of the convex function (2):

$$m(\alpha) = \frac{\{f^{(\alpha)}(1)\}'}{\{f^{(\alpha)}(1)\}''}. \tag{15}$$

Therefore, the α-information matrix $M_X^{(\alpha)}(\psi)$ is positive definite for $\{f^{(\alpha)}(1)\}' > 0$, i.e., for $-\infty < \alpha < 1$.

Let us see how the α-information matrix is related to the Cramér-Rao bound. The inverse of the Fisher information matrix $F_X^{-1}(\psi)$ controls the speed and accuracy of convergence of the usual EM algorithm. We shall see that a similar property applies to the α-EM algorithm. If we use $\alpha \neq -1$, then faster convergence will be obtained for appropriate $m(\alpha)$ [8].

III. α-EM Algorithm and Its Extensions

A. α-Log-Likelihood Ratios for Complete and Incomplete Data

Let $p_{Y|\psi}(y|\psi)$ be a probability density for the observed data vector y in $\mathcal{Y}$ parameterized by ψ, and assume that the regularity conditions (a) $\sim$ (d) apply. The set $\mathcal{Y}$ is viewed as an incomplete data set. Let $x \in \mathcal{X}$ be complete or augmented data which contains unknown, ideal observations. Then, the observed incomplete data y comes from $\mathcal{X}(y) = \{x \mid y(x) = y\} \subseteq \mathcal{X}$. We have [1]

$$p_{Y|\psi}(y|\psi) = \int_{\mathcal{X}(y)} p_{X|\psi}(x|\psi)dx. \tag{16}$$

We assume that the conditional probability density

$$p_{X|Y,\psi}(x|y,\psi) = \frac{p_{X|\psi}(x|\psi)}{p_{Y|\psi}(y|\psi)} \tag{17}$$

is regular. The incomplete-data α-log-likelihood ratio is

$$L_Y^{(\alpha)}(\psi|\varphi) \stackrel{\text{def}}{=} L^{(\alpha)}\left(\frac{p_{Y|\psi}(y|\psi)}{p_{Y|\varphi}(y|\varphi)}\right) = \frac{2}{1+\alpha}\left\{R_Y^{(\alpha)}(\psi|\varphi) - 1\right\}, \tag{18}$$

where

$$R_Y^{(\alpha)}(\psi|\varphi) = \left\{\frac{p_{Y|\psi}(y|\psi)}{p_{Y|\varphi}(y|\varphi)}\right\}^{\frac{1+\alpha}{2}}. \tag{19}$$

On the other hand, the complete-data α-log-likelihood ratio is

$$L_X^{(\alpha)}(\psi|\varphi) \stackrel{\text{def}}{=} L^{(\alpha)}\left(\frac{p_{X|\psi}(x|\psi)}{p_{X|\varphi}(x|\varphi)}\right) = \frac{2}{1+\alpha}\left\{R_X^{(\alpha)}(\psi|\varphi) - 1\right\}, \tag{20}$$

where

$$R_X^{(\alpha)}(\psi|\varphi) = \left\{\frac{p_{X|\psi}(x|\psi)}{p_{X|\varphi}(x|\varphi)}\right\}^{\frac{1+\alpha}{2}}. \tag{21}$$

[5] Conditions on differentiation up to twice are placed in (b) and (c) so that the second equality in (10) holds.

[6] There are many versions of sufficient conditions for the commutativity. One example is as follows (Theorem 2.27 of [23]): $g(x) \in \mathcal{L}_1(\mu)$ exists which upper-bounds the absolute values of the first and second derivatives.

[7] Another type of α-information matrix is

$$\begin{aligned}M_{\exp}^{(\alpha)}(\psi) &\stackrel{\text{def}}{=} E_{\exp(L^{(\alpha)})}[(\partial L^{(\alpha)}/\partial\psi)(\partial L^{(\alpha)}/\partial\psi^T)] \\ &= -E_{\exp(L^{(\alpha)})}[\partial^2 L^{(\alpha)}/\partial\psi\partial\psi^T].\end{aligned}$$

Here, the expectation is taken with respect to

$$\exp\{L^{(\alpha)}(p)\} = \exp\{\tfrac{2}{1+\alpha}(p^{\frac{1+\alpha}{2}} - 1)\}.$$

[8] The inverse of the scale factor $m^{-1}(\alpha) = \frac{2}{1-\alpha}$ can be called the aptitude number in learning theory parlance.

B. Derivation of the α-EM Algorithm

This section derives an extended EM algorithm using the properties of the α-logarithm and the α-divergence. Let Ψ be a set of interest for the generic parameters. Ψ is either $\mathcal{R}$ or a subset thereof. Let $\varphi \in \Psi$ and $\psi \in \Psi$. Then,

$$Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) \stackrel{\text{def}}{=} E_{p_{X|Y},\varphi}\left[L^{(\alpha)}_X(\psi|\varphi)\right] \tag{22}$$

is the conditional expectation of the complete-data α-log-likelihood ratio given the data y and a tentative model φ. Clearly, $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\psi) = Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) = 0$ for every α. In addition, $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ is assumed to satisfy the following properties for α in a finite open interval in $(-\infty, 1)$ including -1 [9]:

(e) $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ exists for all pairs of (ψ, φ).

(f) $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ is a function of class C^∞ with respect to ψ and φ [10]. The differentiation and expectation operators commute [11].

Next, we compute the α-divergence $D^{(\alpha)}_{X|Y}(\varphi\|\psi)$ between two conditional probability densities $p_{X|Y,\varphi}(x|y,\varphi)$ and $p_{X|Y,\psi}(x|y,\psi)$. Then, one obtains the following equation involving the α-divergence by separating the conditional probabilities using Bayes' formula:

$$\begin{aligned}\tfrac{4}{1-\alpha^2}\left\{\tfrac{p_{Y|\psi}(y|\psi)}{p_{Y|\varphi}(y|\varphi)}\right\}^{\frac{1+\alpha}{2}} = \\ \tfrac{4}{1-\alpha^2}\int_{\mathcal{X}} p_{X|Y,\varphi}(x|y,\varphi)\left\{\tfrac{p_{X|\psi}(x|\psi)}{p_{X|\varphi}(x|\varphi)}\right\}^{\frac{1+\alpha}{2}} dx \\ + \left\{\tfrac{p_{Y|\psi}(y|\psi)}{p_{Y|\varphi}(y|\varphi)}\right\}^{\frac{1+\alpha}{2}} D^{(\alpha)}_{X|Y}(\varphi\|\psi).\end{aligned} \tag{23}$$

Let

$$\begin{aligned}S^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) &\stackrel{\text{def}}{=} \int_{\mathcal{X}} p_{X|Y,\varphi}(x|y,\varphi)\left\{\tfrac{p_{X|\psi}(x|\psi)}{p_{X|\varphi}(x|\varphi)}\right\}^{\frac{1+\alpha}{2}} dx \\ &\stackrel{\text{def}}{=} E_{p_{X|Y,\varphi}}\left[\left\{\tfrac{p_{X|\psi}}{p_{X|\varphi}}\right\}^{\frac{1+\alpha}{2}}\right] \\ &= E_{p_{X|Y,\varphi}}\left[R^{(\alpha)}_X(\psi|\varphi)\right].\end{aligned} \tag{24}$$

Then,

$$Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) = \tfrac{2}{1+\alpha}\{S^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) - 1\}. \tag{25}$$

Therefore, from (23) one obtains

$$\begin{aligned}L^{(\alpha)}_Y(\psi|\varphi) = Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) \\ + \tfrac{1-\alpha}{2}\left\{\tfrac{p_{Y|\psi}(y|\psi)}{p_{Y|\varphi}(y|\varphi)}\right\}^{\frac{1+\alpha}{2}} D^{(\alpha)}_{X|Y}(\varphi\|\psi).\end{aligned} \tag{26}$$

This equality is the core of the α-EM algorithm. The case of $\alpha = -1$ is equivalent to Equation (3.10) of [1]. The role of Equation (26) is understood in the following way. First, we check to see what the right-hand side implies. Its second term is nonnegative for $\alpha < 1$. Then, we consider the maximization of the first term $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ with respect to the argument ψ. Under the assumptions specified in the theorems in the next section, a maximum exists for a particular vector, say ψ^*. From Equations (7), (11) and (22), the maximum is positive if $\psi^* \neq \varphi$. In this case, the left-hand side of Equation (26), $L^{(\alpha)}_Y(\psi^*|\varphi)$, is also positive. From Equations (7) and (18), the positivity of $L^{(\alpha)}_Y(\psi^*|\varphi)$ means that $p_{Y|\psi^*}(y|\psi^*) > p_{Y|\varphi}(y|\varphi)$, which indicates that ψ^* is better than φ in the sense of maximum likelihood estimation. Therefore, for the next iteration, we replace the old parameter φ with the newly obtained ψ^*.

Form the above argument, we have the basic α-EM algorithm, of which there are two versions.

[α-EM Algorithm; Version I]

This version is a series of applications of the E-step and the M-step followed by the U-step (update step) for $\alpha < 1$.

[Initialization] The parameter φ is set for the first cycle.

[E-step] Compute $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$. This corresponds to the E-step of the traditional EM-algorithm.

[M-step] Compute $\psi^* = \arg\max_\psi Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$. (27)

[U-step] Replace φ by ψ^* and go back to the E-step until convergence is achieved.

Classification by α leads to the following version, also obtained from Equation (26).

[α-EM Algorithm; Version II]

[Initialization] Inititalize φ.

[E-step] Compute $S^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ in (24). Again this corresponds to the E-step of the traditional EM algorithm.

[M-step] Compute ψ^* as follows.

1) If $-1 < \alpha < 1$:

$$\psi^* = \arg\max_\psi S^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi).$$

2) If $\alpha = -1$:

$$\psi^* = \arg\max_\psi E_{p_{X|Y,\varphi}}\left[\log p_{X|\psi}\right].$$

3) If $\alpha < -1$:

$$\psi^* = \arg\min_\psi S^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi).$$

[U-step] Replace φ by ψ^* and go back to the E-step until convergence is achieved.

The above algorithms assure that the incomplete-data likelihood is increased for $\psi \neq \varphi$. But, this is only a necessary condition for the convergence of ψ to a stationary point in Ψ. Additional conditions are necessary to satisfy global convergence. The next section will discuss this issue together with statements of more extended algorithms.

[9] The log-EM algorithm of [1] states these assumptions for $\alpha = -1$.

[10] Some properties, e.g., Lemma 4 of Section III.C requires differentiability only up to twice.

[11] See Footnote 6.

C. Extensions of the α-EM Algorithm and Convergence Conditions

The exact "arg max" for the α-EM (even for the log-EM) is often difficult to obtain. Therefore, by analogy with the Generalized EM (GEM) algorithm [1], we define a new method, called α-GEM, which allows repeated suboptimal maximizations. The algorithm is described as follows.

[α-GEM Algorithm]

[M-step] For $\alpha < 1$, choose ψ^+ such that

$$Q^{(\alpha)}_{X|Y,\varphi}(\psi^+|\varphi) \geq 0. \tag{28}$$

This α-GEM algorithm includes the α-EM algorithm as a special case since the maximizer of $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ with respect to ψ satisfies (28). The following theorem holds for the α-GEM algorithm.

Theorem 2: Let the mapping $\mathcal{A}^{(\alpha)}$ be an update operation of the α-GEM algorithm, i.e., $\varphi[n+1] = \mathcal{A}^{(\alpha)}(\varphi[n]) \in \Psi$. Then, the incomplete-data α-log-likelihood ratio $L^{(\alpha)}_Y(\varphi[n]|\varphi[0])$ monotonically increases with respect to $\varphi[n]$.

Proof: Since α-GEM generates $Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n]|\varphi[n-1]) \geq 0$, one obtains $L^{(\alpha)}_Y(\varphi[n]|\varphi[n-1]) \geq 0$ from (26). Then,

$$L^{(\alpha)}_Y(\varphi[n]|\varphi[0]) \geq L^{(\alpha)}_Y(\varphi[n-1]|\varphi[0]), \quad n = 1, 2, \ldots$$

by virtue of Lemma 1 (vii). Note that the equalities on $L^{(\alpha)}_Y$ can be removed if $Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n]|\varphi[n-1]) > 0$. □

Next, we consider pointwise convergence. Let $\mathcal{S}$ be the set of stationary points for $\mathcal{A}^{(\alpha)}$ in Ψ. Then, we have the following theorem on global convergence to a stationary point. This theorem is adapted from [24], which presents useful sufficient conditions for the well-known global convergence theorem of nonlinear programming [25], [26]. For the α-GEM, conditions are placed on the α-log likelihood ratio and the $Q^{(\alpha)}$-function.

Theorem 3: Assume the following conditions (a) $\sim$ (e) on the α-GEM mapping $\mathcal{A}^{(\alpha)}$ with $\alpha < 1$:

(a) $L^{(\alpha)}_Y(\psi|\varphi)$ is continuous in $\Psi \times \Psi$ and differentiable in the interior of $\Psi \times \Psi$.
(b) $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ is continuous in $\Psi \times \Psi$ and differentiable in the interior of $\Psi \times \Psi$.
(c) The set $\Psi_\varphi = \{\theta \in \Psi \mid L^{(\alpha)}_Y(\theta|\varphi) \geq 0\}$ is compact for any $p_{Y|\varphi}(y|\varphi) > 0$ and $\varphi \in \Psi$.
(d) The set Ψ_φ is in the interior of Ψ.
(e) $L^{(\alpha)}_Y(\varphi[n+1]|\varphi[n]) > 0$ for all $\varphi[n] \in \Psi \backslash \mathcal{S}$ and n.

Then, the following holds.

(i) All limit points of $\{\varphi[n]\}$ are stationary points of $L^{(\alpha)}_Y(\psi|\varphi[0])$ with respect to ψ.
(ii) $L^{(\alpha)}_Y(\varphi[n]|\varphi[0])$ converges monotonically to $L^{(\alpha)}_Y(\varphi^*|\varphi[0])$ for some $\varphi^* \in \mathcal{S}$.

The proof is given in Appendix A. Note that (e) is satisfied if $Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+1]|\varphi[n]) > 0$ for $\varphi[n] \in \Psi \backslash \mathcal{S}$ because of (26).

If we replace the set $\mathcal{S}$ of condition (e) with $\mathcal{M}$, which stands for the set of local maxima in the interior of Ψ, then the following (i) and (ii) hold.

(i) All limit points of $\{\varphi[n]\}$ are local maxima of $L^{(\alpha)}_Y(\psi|\varphi[0])$ with respect to ψ.
(ii) $L^{(\alpha)}_Y(\varphi[n]|\varphi[0])$ converges monotonically to $L^{(\alpha)}_Y(\varphi^*|\varphi[0])$ for some $\varphi^* \in \mathcal{M}$.

In the α-GEM (and α-EM) algorithm, differentiation is often a powerful tool for finding update values. The following lemma relates to the stationary points of $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ and $L^{(\alpha)}_Y(\psi|\varphi)$.

Lemma 4: Assume that $L^{(\alpha)}_Y(\psi|\varphi)$ and $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ are twice differentiable with respect to ψ and differentiable with respect to φ. Expectation and differentiation are assumed to be commutative. Write

$$\partial^{ij} f(\varphi|\varphi) \stackrel{\text{def}}{=} \left.\frac{\partial^{i+j} f(\psi|\varphi)}{\partial^i \psi \partial^j \varphi}\right|_{\psi=\varphi}.$$

Then, the following equalities hold:

$$\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) = \partial^{10} L^{(\alpha)}_Y(\varphi|\varphi) = \frac{\partial \log p(y|\varphi)}{\partial \varphi} \stackrel{\text{def}}{=} \partial \log p(y|\varphi) \stackrel{\text{def}}{=} \partial \ell_Y(\varphi), \tag{29}$$

$$\partial^{11} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) = \partial^{11} L^{(\alpha)}_Y(\varphi|\varphi) + \tfrac{1-\alpha}{2} F_{X|Y}(\varphi), \tag{30}$$

$$-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) = \int_{\mathcal{X}} p(x|y,\varphi) \left\{ \left(-\frac{\partial^2 \log p(x|\varphi)}{\partial \varphi \partial \varphi^T} \right) - \tfrac{1+\alpha}{2} \frac{\partial \log p(x|\varphi)}{\partial \varphi} \frac{\partial \log p(x|\varphi)}{\partial \varphi^T} \right\} dx, \tag{31}$$

$$\partial^{11} L^{(\alpha)}_Y(\varphi|\varphi) = -\tfrac{1+\alpha}{2} \frac{\partial \log p(y|\varphi)}{\partial \varphi} \frac{\partial \log p(y|\varphi)}{\partial \varphi^T}, \tag{32}$$

$$-\partial^{20} L^{(\alpha)}_Y(\varphi|\varphi) = \left(-\frac{\partial^2 \log p(y|\varphi)}{\partial \varphi \partial \varphi^T} \right) - \tfrac{1+\alpha}{2} \frac{\partial \log p(y|\varphi)}{\partial \varphi} \frac{\partial \log p(y|\varphi)}{\partial \varphi^T}, \tag{33}$$

$$\partial^{10} D^{(\alpha)}_{X|Y}(\varphi|\varphi) = 0, \tag{34}$$

$$\partial^{20} D^{(\alpha)}_{X|Y}(\varphi|\varphi) = F_{X|Y}(\varphi). \tag{35}$$

Here, $F_{X|Y}(\varphi)$ is the Fisher information matrix with respect to the conditional probability density $p(x|y,\varphi)$.

All equalities can be obtained directly using (3), (6) and (26). Note that (29) means that differentiable stationary points are not affected by the choice of α.

If the parameter vector φ is viewed as a collection of subvectors, then each subvector can be updated one by one. Such an idea leads to our α-version of the ECM algorithm (Expectation/Conditional Maximization [27]). The above component grouping can be understood as a special case of the following constraint.

Let

$$\mathcal{G} \stackrel{\text{def}}{=} \{g_s(\theta) \in \boldsymbol{R}^{d_s},\ s = 1, \ldots, S \mid \theta \in \boldsymbol{R}^d\}$$

be a set of d_s-dimensional vector functions used for constraints. Define

$$\Psi_s(\varphi) \stackrel{\text{def}}{=} \{\theta \in \Psi \mid g_s(\theta) = g_s(\varphi)\} \subset \Psi \subset \boldsymbol{R}^d,\ s = 1, \ldots, S.$$

Then, the α-version of the ECM algorithm is described as follows.

[α-ECM Algorithm]

[Initialization] Choose $\varphi[0]$.

[E-step at cycle $n+1$] This is the same as the α-EM algorithm. Here, $\varphi[n]$ is already given at the previous cycle.

[CM-step at cycle $n+1$] The following substeps are performed for $s = 1, \ldots, S$.

[Maximization substep at $(n+s/S)$] Choose a maximizer $\varphi[n+s/S] \in \Psi_s(\varphi[n+(s-1)/S])$ such that

$$Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+s/S]|\varphi[n]) \geq 0. \quad (36)$$

[U-Step at cycle $n+1$] Update $\varphi[n]$ by $\varphi[n+S/S]$, i.e., by $\varphi[n+1]$. Then, go back to the E-step until convergence is achieved.

Since $\varphi[n+s/S] \in \Psi_s(\varphi[n+(s-1)/S])$ for $s = 1, \ldots, S$, one obtains the case of $s = S$ such that $Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+1]|\varphi[n]) \geq 0$ from the inequality (36). Thus, α-ECM is a special case of α-GEM. Therefore, Theorems 2 and 3 hold for α-ECM. Hence, the α-ECM algorithm can give stationary points. But, these might be stationary points only for a specific subspace of Ψ. Thus, it is desirable to pose appropriate conditions on $\mathcal{G}$ which guarantee that α-ECM limit points are unconstrained ones. This is the space-filling condition given in [27]. We list the main properties of this space-filling condition which will be used for our α-ECM convergence.

The set of constraint functions $\mathcal{G}$ is said to be space-filling at φ if the set of tangent cones at φ, say

$$\{T_s(\varphi) \mid s = 1, \ldots, S\},$$

spans the whole $\boldsymbol{R}^d$. Here, the s-th tangent cone at φ is defined by

$$T_s(\varphi) \stackrel{\text{def}}{=} \left\{\eta_s \in \boldsymbol{R}^d \mid \exists\{\varphi_i\} \in \Psi_s(\varphi) \text{ such that } \eta_s = \lim_{i\to\infty} \frac{\varphi_i - \varphi}{\|\varphi_i - \varphi\|}\right\} \subset \boldsymbol{R}^d. \quad (37)$$

Then, the space-filling condition on $\mathcal{G}$ at φ means that

$$T(\varphi) \stackrel{\text{def}}{=} \text{closure}\{\textstyle\sum_{s=1}^{S} a_s\eta_s \mid a_s \geq 0,\ \eta_s \in T_s(\varphi)\} = \boldsymbol{R}^d. \quad (38)$$

If $g_s(\varphi)$, $(s = 1, \ldots, S)$, are differentiable and the gradient $\nabla g_s(\varphi)$ is full rank at $\varphi \in \{\text{interior of } \Psi\}$, then "space-filling at φ" is equivalent to

$$J(\varphi) \stackrel{\text{def}}{=} \{\xi \mid \xi^T\eta \leq 0 \text{ for } \forall\eta \in T(\varphi)\} \quad (39)$$

$$= \textstyle\bigcap_{s=1}^{S} J_s(\varphi) = \mathbf{0}. \quad (40)$$

Here, $J_s(\varphi)$ is the column space of the gradient:

$$J_s(\varphi) = \{\nabla g_s(\varphi)\lambda_s \mid \lambda_s \in \boldsymbol{R}^{d_s}\}.$$

Theorem 5: Assume the conditions (a) $\sim$ (d) of Theorem 3. Also, assume the following (e) $\sim$ (g).

(e) $\varphi[n+s/S]$ in (36) is the unique maximizer in $\Psi_s(\varphi[n+(s-1)/S])$.

(f) $\mathcal{G}$ is space-filling at $\varphi = \varphi[n]$ with full rank $\nabla g_s(\varphi)$, $s = 1, \ldots, S$, for each iteration $n > 0$.

(g) The incomplete-data α-log-likelihood ratio satisfies

$$L^{(\alpha)}_Y(\varphi[n+1]|\varphi[n]) > 0 \text{ for all } \varphi[n] \in \Psi\backslash\mathcal{S}.$$

Here, $\mathcal{S}$ is the set of stationary points of the α-ECM algorithm.

Then, the following (i) and (ii) hold:

(i) All limit points of the α-ECM sequences are unconstrained stationary points of $L^{(\alpha)}_Y(\psi|\varphi[0])$ with respect to ψ.

(ii) $L^{(\alpha)}_Y(\varphi[n]|\varphi[0])$ converges monotonically to $L^{(\alpha)}_Y(\varphi^*|\varphi[0])$ for some $\varphi^* \in \mathcal{S}$.

The proof using Lemmas 1 and 4 is given in Appendix B. Note that (g) is satisfied if $Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+1]|\varphi[n]) > 0$ for $\varphi[n] \in \Psi\backslash\mathcal{S}$.

In closing this section, we give variants of α-ECM in parallel to [27]. If we change the CM-step maximizer of (36) to

$$Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+s/S]|\varphi[n+(s-1)/S]) \geq 0, \quad (41)$$

then the strategy becomes a multicycle α-ECM algorithm. If we allow weaker maximizations, e.g., a series of suboptimal maximizations satisfying (41), the algorithm becomes a multicycle α-GEM. The multicycle α-GEM maintains the monotone increasing property of the incomplete-data α-log-likelihood ratio.

D. α-ECME and Further Generalization

As explained in the previous section, the α-ECM algorithm indirectly keeps the incomplete-data's increase by the use of (36). If direct maximization of (18) is possible for some CM steps, the total M-step becomes more efficient. This is the idea of ECME (Expectation/Conditional Maximization Either) [28]. Its α-version is as follows.

[α-ECME Algorithm]

[Initialization] Choose $\varphi[0]$.

[E-step at $n+1$] This is the same as in α-EM. Here, $\varphi[n]$ is already given at the previous cycle.

[CM-step at $n+1$] The following CM-$Q^{(\alpha)}$-substeps are performed for $s = 1, \ldots, S_c$.

[CM-$Q^{(\alpha)}$-substep at $(n+s/S)$] Choose a maximizer $\varphi[n+s/S] \in \Psi_s(\varphi[n+(s-1)/S])$ such that

$$Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+s/S]|\varphi[n]) \geq 0. \quad (42)$$

For $s = S_c+1, \ldots, S$, the following CM-$L^{(\alpha)}_Y$-substeps are performed.

[CM-$L^{(\alpha)}_Y$-substep at $(n+s/S)$] Choose a maximizer $\varphi[n+s/S] \in \Psi_s(\varphi[n+(s-1)/S])$ such that

$$L^{(\alpha)}_Y(\varphi[n+s/S]|\varphi[n]) \geq 0. \quad (43)$$

[U-Step at cycle $n+1$] Update $\varphi[n]$ by $\varphi[n+S/S]$, i.e., by $\varphi[n+1]$.

Convergence can be discussed similarly to [28] with modifications adapted to (42) and (43).

The α-ECME interleaves the direct maximization of (43) following the $Q^{(\alpha)}$-maximization of (42). This is equivalent to the spacer of nonlinear programming [26]. Therefore, each of the substeps can be repeated or skipped. This idea of cycle control is adopted in the AECM (Alternating Expectation-Conditional Maximization) algorithm [5] which is a generalized unification of ECM and SAGE (Space-Alternating Generalized Expectation-Maximization [29]). Since the data augmentation using a working parameter, which is another important artifice of AECM, is independent of the α-logarithmic strategy, the α-AECM algorithm can be obtained [12]. Therefore, we let *α-EM family* be the generic name for the α-{EM, GEM, ECM, ECME, SAGE, AECM} algorithms, the multicycle α-{GEM, ECM, ECME}, and so forth. The terminology "log-EM family" is reserved for the case of $\alpha = -1$.

IV. Convergence Speed

Here, we evaluate the convergence speed of the α-EM family using Lemma 4. Speed evaluation by experiments on specific examples will be given in Section V.

A. Effects of $Q^{(\alpha)}$ and $L_Y^{(\alpha)}$

First, we evaluate the second-order properties of $Q^{(\alpha)}$ and $L_Y^{(\alpha)}$. Assume that $\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ can be expanded in a Taylor series with respect to ψ. Here, φ need not be a stationary point φ^* of the α-EM family such that $\mathcal{A}^{(\alpha)}(\varphi^*) = (\varphi^*)$. In the neighborhood of φ, we have

$$\begin{aligned}\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) &= \partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) \\ &\quad + \partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi)(\psi-\varphi) + o(1).\end{aligned}$$

Since (29) holds, the M-step gives

$$\begin{aligned}0 &= \partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\psi^*|\varphi) \\ &= \partial \ell_Y(\varphi) + \partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi)(\psi^*-\varphi) + o(1). \quad (44)\end{aligned}$$

Here, ψ^* is a vector satisfying the left equality of (44) in the neighborhood of φ. $o(1)$ stands for higher-order terms. Then, we have the following theorem.

Theorem 6: Suppose $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ can be expanded in a Taylor series with respect to ψ. Then, each M-step towards ψ^* of (44), which is in the neighborhood of φ, satisfies

$$\psi^* = \varphi + \left[-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi)\right]^{-1} \partial \ell_Y(\varphi)\{1+o(1)\}. \quad (45)$$

[12]Description of the α-AECM indexing system is omitted since it requires much notational preparation for *step*, *cycle*, and *iteration* on (36), (41) and (43). The data augmentation and the space filling condition can depend on the cycle index. We refer readers to Sections 3.2 and 3.3 of [5] for these details.

Thus, the α-EM algorithms exploit the second-order property of (31). In case of the α-ECME which also uses the second order property of (33), we have

$$\psi^* = \varphi + \left[-\partial^{20} L_Y^{(\alpha)}(\varphi|\varphi)\right]^{-1} \partial \ell_Y(\varphi)\{1+o(1)\}. \quad (46)$$

The update matrices in (45) and (46) will be evaluated in the next section.

Next, we evaluate the convergence of parameter vectors in the neighborhood of the stationary point φ^*. Let $\partial^{10} Q^{(\alpha)}(\psi|\varphi)$ be a function of a pair of variables (ψ, φ). Suppose this can be expanded in a Taylor series at (φ^*, φ^*) so that

$$\begin{aligned}\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi) &= \partial^{10} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*) \\ &\quad + \partial^{20} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)(\psi-\varphi^*) \\ &\quad + \partial^{11} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)(\varphi-\varphi^*) + o(1). \quad (47)\end{aligned}$$

If we use differentiation at each M-step in order to obtain the update value ψ^*, then the left-hand side with $\psi = \psi^*$ is equal to zero. In addition, the first term of the right-hand side is zero because of Lemma 4 and $\partial \log p(y|\varphi^*) = 0$. Then, we have

$$\begin{aligned}\psi^* - \varphi^* &= \left[-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)\right]^{-1} \\ &\quad \times \left[\partial^{11} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)\right](\varphi-\varphi^*) + o(1). \quad (48)\end{aligned}$$

This means that the Jacobian matrix at the stationary point φ^* is

$$\begin{aligned}J^{(\alpha)}(\varphi^*) &= \left[-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)\right]^{-1} \\ &\quad \times \left[\partial^{11} Q^{(\alpha)}_{X|Y,\varphi^*}(\varphi^*|\varphi^*)\right]. \quad (49)\end{aligned}$$

Therefore, we have the following theorem.

Theorem 7: Suppose $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi)$ can be expanded in a Taylor series with respect to (ψ, φ) at (φ^*, φ^*), where φ^* is a stationary point of the α-EM algorithm. Then,

$$\psi^* - \varphi^* = J^{(\alpha)}(\varphi^*)(\varphi-\varphi^*) + o(1). \quad (50)$$

Here, ψ^* is an M-step's update value satisfying

$$\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\psi^*|\varphi) = 0.$$

Due to (50), a smaller Jacobian matrix will result in faster convergence around the stationary point φ^*. The Jacobian matrix $J^{(\alpha)}$ will be evaluated in the next section.

B. Evaluation of Update Matrices

Theorems 6 and 7, with the aid of Lemma 4, suggest that one may choose $\alpha \neq -1$ to improve the convergence speed of the log-EM family. First, we discuss (45) in the case of the exponential family.

Let $p_{X|\varphi}(x|\varphi)$ be a class of exponential family such that

$$p_{X|\varphi}(x|\varphi) = \exp\left\{\varphi^T r_X(x) - k_X(x) - \theta_X(\varphi)\right\}. \quad (51)$$

Then, we have from (31) that

$$\begin{aligned}
&-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) \\
&= \tfrac{\partial^2 \theta_X(\varphi)}{\partial\varphi\partial\varphi^T} - \tfrac{1+\alpha}{2}\int_{\mathcal{X}} \left\{ r_X(x) - \tfrac{\partial \theta_X(\varphi)}{\partial\varphi} \right\} \\
&\qquad \times \left\{ r_X(x) - \tfrac{\partial \theta_X(\varphi)}{\partial\varphi} \right\}^T p(x|y,\varphi)dx \\
&\stackrel{\text{def}}{=} F_X(\varphi) - \tfrac{1+\alpha}{2} N_{X|Y}(\varphi) \\
&= F_X(\varphi)\left[I - \tfrac{1+\alpha}{2} F_X^{-1}(\varphi) N_{X|Y}(\varphi) \right]. \qquad (52)
\end{aligned}$$

Here, $F_X(\varphi)$ is the Fisher information matrix for $p_{X|\varphi}(x|\varphi)$. Since $F_X(\varphi)$ and $N_{X|Y}(\varphi)$ are positive definite, a number β_Q exists such that $-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) > 0$ holds for $\alpha < \beta_Q$. Therefore, the update (45) converges for $\alpha < \min(\beta_Q, 1) \stackrel{\text{def}}{=} \beta_Q^*$. In addition, we have the following inequality on the spectral radius for $-1 < \alpha < \beta_Q^*$:

$$\rho\left(\left[I - \tfrac{1+\alpha}{2} F_X^{-1}(\varphi) N_{X|Y}(\varphi) \right]^{-1} \right) > 1.$$

Therefore, we have the following corollary to Theorem 6.

Corollary 8: For the α-EM algorithm on the exponential family, the following inequality on the spectral radii holds for $-1 < \alpha < \beta_Q^*$:

$$\rho\left(-\{\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi)\}^{-1} \right) > \rho\left(-\{\partial^{20} Q^{(-1)}_{X|Y,\varphi}(\varphi|\varphi)\}^{-1} \right). \qquad (53)$$

We note here that the right-hand side of (53) is the case of $\alpha = -1$, i.e., the log-EM family. Thus, the speedup of the α-EM algorithm is based on the eigenvalue adjustment of the update matrix by virtue of α as is expressed in (52).

For the α-ECME algorithm which exploits the second-order property of (33), a discussion parallel to the above is possible. Let the observed data come from an exponential family such that

$$p_{Y|\varphi}(y|\varphi) = \exp\left\{ \varphi^T r_Y(y) - k_Y(y) - \theta_Y(\varphi) \right\}. \qquad (54)$$

Then, for $-1 < \alpha < \beta_L^*$, where β_L^* is defined similarly to β_Q^* above, we have

$$\rho\left(\left[I - \tfrac{1+\alpha}{2} F_Y^{-1}(\varphi) N_Y(\varphi) \right]^{-1} \right) > 1.$$

Here, $F_Y(\varphi)$ is the Fisher information matrix for $p_{Y|\varphi}(y|\varphi)$, and

$$N_Y(\varphi) = \left\{ r_Y(y) - \tfrac{\partial \theta_Y(\varphi)}{\partial\varphi} \right\} \left\{ r_Y(y) - \tfrac{\partial \theta_Y(\varphi)}{\partial\varphi} \right\}^T.$$

Then, we have

$$\rho\left(-\{\partial^{20} L_Y^{(\alpha)}(\varphi|\varphi)\}^{-1} \right) > \rho\left(-\{\partial^{20} L_Y^{(-1)}(\varphi|\varphi)\}^{-1} \right)$$

for $-1 < \alpha < \beta_L^*$.

Next, we discuss the Jacobian matrix of (49). At the stationary point φ^*, we have

$$\begin{aligned}
\left[-\partial^{20} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) \right]_{\varphi=\varphi^*} &= \tfrac{1-\alpha}{2} F_{X|Y}(\varphi^*) \\
&\quad + E_{p(x|y,\varphi^*)}\left[-\partial^2 \log p(y|\varphi^*) \right]
\end{aligned}$$

and

$$\left[\partial^{11} Q^{(\alpha)}_{X|Y,\varphi}(\varphi|\varphi) \right]_{\varphi=\varphi^*} = \tfrac{1-\alpha}{2} F_{X|Y}(\varphi^*).$$

from Lemma 4. Here, $F_{X|Y}(\varphi^*)$ is the Fisher information matrix with respect to $p_{X|Y,\varphi^*}(x|y,\varphi^*)$. Therefore, we have the following corollary to Theorem 7.

Corollary 9: If $E_{p(x|y,\varphi^*)}\left[-\partial^2 \log p(y|\varphi^*) \right] > 0$, then $J^{(\alpha)}(\varphi^*) < J^{(-1)}(\varphi^*)$ holds for $\alpha \in (-1, 1)$.

Note that a sufficient condition for Corollary 9 is $-\partial^2 \log p(y|\varphi^*) > 0$. This inequality is satisfied if the incomplete data comes from an exponential family, such as (54).

V. Examples on Gaussian Mixtures

This section experimentally verifies the theoretically-obtained speedup of Section IV [13]. Two different types of Gaussian mixtures are chosen as benchmarks. These problems are examples of exponential families.

A. Clustering for Gaussian Mixtures

Suppose we collect N independent m-dimensional samples $\{x(t)\}_{t=1}^N$. Each sample is assumed to come from one of K Gaussian probability densities

$$p_{X|\Psi}(x(t)|\psi_i) = \mathcal{N}(\mu_{\psi_i}, \mathbf{\Sigma}_{\psi_i}), \quad (i = 1, \ldots, K). \qquad (55)$$

Here, μ_{ψ_i} is a mean vector and $\mathbf{\Sigma}_{\psi_i}$ is a covariance matrix. The i-th Gaussian probability density is selected with probability

$$p_{Z|X,\Psi}(Z(t) = i|x(t), \psi_i) \stackrel{\text{def}}{=} \gamma_{\psi_i}, \quad (i = 1, \ldots, K). \qquad (56)$$

Note that $\sum_{i=1}^K \gamma_{\psi_i} = 1$. Therefore, the set of the parameters to be estimated is $\{(\gamma_{\psi_i}, \mu_{\psi_i}, \mathbf{\Sigma}_{\psi_i})\}_{i=1}^K$. This is the problem of Gaussian mixtures [14] [1], [30], [31]. Table I gives a notational correspondence to the α-EM family of Section III.

TABLE I is Here.

The complete-data probability density is

$$\mathcal{P}_{\mathcal{C}}(\mathcal{X}, \mathcal{Z}|\Psi) = \prod_{t=1}^N \prod_{i=1}^K \left\{ \gamma_{\psi_i} p_{X|\Psi}(x(t)|\psi_i) \right\}^{I_i(Z(t))}.$$

Here, I_i is the indicator function for $Z(t) = i$. Then, the incomplete-data probability density is

$$\mathcal{P}_{\mathcal{I}}(\mathcal{X}|\Psi) = \prod_{t=1}^N \sum_{i=1}^K \left\{ \gamma_{\psi_i} p_{X|\Psi}(x(t)|\psi_i) \right\}. \qquad (57)$$

The complete-data α-log-likelihood ratio is

$$L_{\mathcal{C}}^{(\alpha)}(\mathcal{X}, \mathcal{Z}|\Psi, \Phi) = \tfrac{2}{1+\alpha}\left[\left\{ \tfrac{\mathcal{P}_{\mathcal{C}}(\mathcal{X}, \mathcal{Z}|\Psi)}{\mathcal{P}_{\mathcal{C}}(\mathcal{X}, \mathcal{Z}|\Phi)} \right\}^{\frac{1+\alpha}{2}} - 1 \right].$$

[13] Gradient ascent methods using the α-efficient score defined by (12) is given in [18].

[14] In learning-theory parlance, this problem is classified as *unsupervised learning*.

Then, by taking the conditional expectation on $L_C^{(\alpha)}$, we obtain the surrogate function or the minorant $Q^{(\alpha)}$:

$$\begin{aligned} Q^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\Phi}(\Psi|\Phi) &= E_{\mathcal{P}_C(\mathcal{Z}|\mathcal{X},\Phi)}\left[L_C^{(\alpha)}(\mathcal{X},\mathcal{Z}|\Psi,\Phi)\right] \\ &= \frac{2}{1+\alpha}\left[S^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\Phi} - 1\right] \end{aligned} \tag{58}$$

Here,

$$\begin{aligned} S^{(\alpha)}_{\mathcal{X},\mathcal{Z}|\Phi} &= \textstyle\prod_{t=1}^N W^{(\alpha)}(t), \\ W^{(\alpha)}(t) &= \textstyle\sum_{i=1}^K h_i^{(\alpha)}(t), \\ h_i^{(\alpha)}(t) &= \left\{\frac{\gamma_{\psi_i} p(x(t)|\psi_i)}{\gamma_{\varphi_i} p(x(t)|\varphi_i)}\right\}^{\frac{1+\alpha}{2}} h_i(t), \end{aligned}$$

and

$$h_i(t) \stackrel{\text{def}}{=} E_{\mathcal{P}_C(\mathcal{Z}|\mathcal{X},\Phi)}\left[I_i(Z(t))\right] = \frac{\gamma_{\varphi_i} p(x(t)|\varphi_i)}{\sum_{j=1}^K \gamma_{\varphi_j} p(x(t)|\varphi_j)}.$$

Define the following normalization:

$$\tilde{h}_i^{(\alpha)}(t) = h_i^{(\alpha)}(t)/W^{(\alpha)}(t). \tag{59}$$

Note that $\sum_{i=1}^K \tilde{h}_i^{(\alpha)}(t) = 1$ holds. Then, we have the following set of update equations.

$$\gamma_{\psi_i} = \frac{1}{N}\textstyle\sum_{t=1}^N \tilde{h}_i^{(\alpha)}(t) \tag{60}$$

$$\mu_{\psi_i} = \{\textstyle\sum_{t=1}^N \tilde{h}_i^{(\alpha)}(t)x(t)\}/\{\sum_{t=1}^N \tilde{h}_i^{(\alpha)}(t)\} \tag{61}$$

$$\begin{aligned} \mathbf{\Sigma}_{\psi_i} = \{&\textstyle\sum_{t=1}^N \tilde{h}_i^{(\alpha)}(t)(x(t)-\mu_{\psi_i})(x(t)-\mu_{\psi_i})^T\} \\ &/\{\textstyle\sum_{t=1}^N \tilde{h}_i^{(\alpha)}(t)\} \end{aligned} \tag{62}$$

Equation (60) can be obtained by differentiating

$$Q^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\Psi} + \lambda(\textstyle\sum_{j=1}^K \gamma_{\psi_j} - 1)$$

by γ_{ψ_i} using the relationship

$$\log S^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\Psi} = \textstyle\sum_{t=1}^N \log W^{(\alpha)}(t).$$

Here, λ is a Lagrange multiplier. The other update equations (61) and (62) can be obtained by differentiating $Q^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\Phi}$ with respect to the update parameters. Note that ψ_i in the right-hand side is one step before that of the left-hand side. Therefore, at the k-th iteration to obtain $\psi_i = \varphi_i[k+1]$, the right-hand sides of (60), (61) and (62) use $\psi_i = \varphi_i[k]$ and $\varphi_i = \varphi_i[k-1]$.

Figure 1 is Here.

Figure 2 is Here.

Figure 1 illustrates a set of source data [32] and the obtained mean vectors for $K = 4$. Figure 2 illustrates convergence curves for the incomplete-data log-likelihood $\log \mathcal{P}_{\mathcal{I}}(\mathcal{X}|\Psi)$ using the α-EM algorithm of Section IV; observe that the α-EM algorithm outperforms the traditional log-EM algorithm.

Table II shows a speedup comparison. The second and third columns show that α-EM is $37/14 = 2.64$ times faster than the log-EM (the case of $\alpha = -1$). Note that the case of "$\alpha = 1.0$ (until the tenth iteration) and $\alpha < 1.0$ (e.g., 0.0) thereafter" gives the fastest rise-up. But, the number of required iterations is the same as in the case of "$\alpha = 0.6$ (constant)." The fourth and fifth columns show a more practical comparison based upon CPU time. The α-EM required 1.44 times more CPU time [15] per iteration. But, the α-EM is still faster than the log-EM by a CPU-time speedup ratio of $(37/14)/1.44 = 1.84$.

TABLE II is Here.

In addition to the improved convergence speed, the α-EM family has another interesting feature. The posterior probabilities $h_i(t)$, $(i = 1,\ldots,N)$, specify a soft-max decision for the data $x(t)$. If such a soft-max decision is replaced by a hard-max decision, this becomes a vector quantization. This algorithm can be made more sophisticated so that it incorporates self-organization [32], [33]. Therefore, vector quantization and its associated self-organization can trace up their hierarchy to the α-EM algorithm. Initial discussions on this issue can be found in [34].

B. Source Estimation by Combined Gaussian Mixtures

Here, we consider another type of Gaussian mixture. In this problem, the output of one group of Gaussian mixtures is controlled by another group of Gaussian mixtures [35]. Therefore, we denote the generic parameter by $\psi = (\theta^\psi, \xi^\psi)$. In the first group of Gaussian mixtures, an output $y(t)$ is produced when $x(t)$ is input. There are K possible outputs according to the following Gaussian probability set:

$$p_{Y|X,\Theta}(y(t)|x(t),\theta_i^\psi) = \mathcal{N}(\mu_i^\psi, \mathbf{\Sigma}_i^\psi), \quad (i = 1,\ldots,K). \tag{63}$$

The input $x(t)$ affects the mean vector by

$$\mu_i^\psi = f_i(x(t),\theta_i^\psi) = \Theta_i^\psi x(t).$$

The sequence of input-output pairs $(x(t),y(t))$ [16] are the observed incomplete data

$$\mathcal{I} = (\mathcal{X},\mathcal{Y}) = (\{x(t)\}_{t=1}^N, \{y(t)\}_{t=1}^N).$$

The missing data is a random variable $Z(t)$ which specifies one of the K Gaussian densities in (63). Thus, the complete data is $\mathcal{C} = (\mathcal{I},\mathcal{Z})$, where $\mathcal{Z} = \{Z(t)\}_{t=1}^N$.

The probability of the missing data is controlled by another group of Gaussian elements [17]

$$p_{X|\Xi}(x(t)|\xi_i^\psi) = \mathcal{N}(m_i^\psi, \Gamma_i^\psi), \quad (i = 1,\ldots,K). \tag{64}$$

The event $\{Z(t) = i\}$ indicates $(\theta_i^\psi, \xi_i^\psi)$. Therefore,

$$\begin{aligned} p(x(t)|\xi^\psi) &= \textstyle\sum_{i=1}^K p(Z(t)=i|\xi^\psi)p(x(t),Z(t)=i|\xi^\psi) \\ &= \textstyle\sum_{i=1}^K \omega_i^\psi p(x(t)|\xi_i^\psi), \end{aligned}$$

where

$$\omega_i^\psi = p(Z(t) = i|\xi_i^\psi).$$

[15] Throughout the examples, CPU time was measured on programs using the `gcc` compiler and `<time.h>` library.

[16] In learning-theory parlance, this problem is called *supervised learning*.

[17] A different type of the mixture selector is discussed in [17], [18] and [36]

Thus, the incomplete-data likelihood to be maximized is

$$\begin{aligned}&\prod_{t=1}^{N} p_{X,Y|\Psi}(x(t),y(t)|\Psi)\\&\quad=\prod_{t=1}^{N} p(x(t)|\xi^{\psi})p(y(t)|x(t),\theta^{\psi})\\&\quad=\prod_{t=1}^{N}\sum_{i=1}^{K}\omega_i^{\psi}p(x(t)|\xi_i^{\psi})p(y(t)|x(t),\theta_i^{\psi}).\end{aligned}$$

The corresponding complete-data likelihood is

$$\begin{aligned}&\mathcal{P}_C(\mathcal{X},\mathcal{Y},\mathcal{Z}|\Psi)\\&\quad=\prod_{t=1}^{N}\prod_{i=1}^{K}\{\omega_i^{\psi}p(x(t)|\xi_i^{\psi})p(y(t)|x(t),\theta_i^{\psi})\}^{I_i(Z(t))}.\end{aligned}$$

Therefore, the surrogate function or the minorant $Q^{(\alpha)}$ to be maximized is

$$\begin{aligned}Q^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi}(\Psi|\Phi) &= E_{\mathcal{P}_C(\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi)}\left[L^{(\alpha)}\left(\frac{\mathcal{P}_C(\mathcal{X},\mathcal{Y},\mathcal{Z}|\Psi)}{\mathcal{P}_C(\mathcal{X},\mathcal{Y},\mathcal{Z}|\Phi)}\right)\right]\\&\stackrel{\text{def}}{=}\frac{2}{1+\alpha}\left\{S^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi}(\Psi|\Phi)-1\right\}. \qquad (65)\end{aligned}$$

Now, we can derive a set of update equations for ω_i^{ψ} and (64). Our methods are the same as in Section V.A. The update equation for ω_i^{ψ} is obtained by differentiating $Q^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi}(\Psi|\Phi)+\lambda(\sum_{i=1}^{N}\omega_i^{\psi}-1)$. Here, λ is a Lagrange multiplier. In this computation, the following relationships are used:

$$\begin{aligned}&S^{(\alpha)}_{\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi}=\prod_{t=1}^{N}W^{(\alpha)}(t),\\&W^{(\alpha)}(t)=\sum_{i=1}^{K}h_i^{(\alpha)}(t),\\&h_i^{(\alpha)}(t)=\left\{\frac{\omega_i^{\psi}p(x(t)|\xi_i^{\psi})}{\omega_i^{\varphi}p(x(t)|\xi_i^{\varphi})}\right\}^{\frac{1+\alpha}{2}}h_i(t),\\&h_i(t)\stackrel{\text{def}}{=}E_{\mathcal{P}_C(\mathcal{Z}|\mathcal{X},\mathcal{Y},\Phi)}[I_i(Z(t))]=\frac{\omega_i^{\varphi}p(x(t)|\xi_i^{\varphi})}{\sum_{j=1}^{K}\omega_j^{\varphi}p(x(t)|\xi_j^{\varphi})}.\end{aligned}$$

Next, we define $\tilde{h}_i^{(\alpha)}(t)$ as in (59). Then, we have the following update equations:

$$\omega_i^{\psi}=\frac{1}{N}\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t), \qquad (66)$$

$$m_i^{\psi}=\{\textstyle\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)x(t)\}/\{\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)\}, \qquad (67)$$

$$\begin{aligned}\Gamma_i^{\psi}=&\{\textstyle\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)(x(t)-m_i^{\psi})(x(t)-m_i^{\psi})^T\}\\&/\{\textstyle\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)\}. \qquad (68)\end{aligned}$$

Next, we derive a set of update equations for (63). In this case, differentiation of the $Q^{(\alpha)}$-function gives

$$\Theta_i^{\psi}=\left\{\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}y(t)x(t)^T\right\}\left\{\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}x(t)x(t)^T\right\}^{-1}, \qquad (69)$$

$$\begin{aligned}\mathbf{\Sigma}_i^{\psi}=&\left\{\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)(y(t)-\Theta_i^{\psi}x(t))(y(t)-\Theta_i^{\psi}x(t))^T\right\}\\&\Big/\left\{\sum_{t=1}^{N}\tilde{h}_i^{(\alpha)}(t)\right\}. \qquad (70)\end{aligned}$$

Similarly to Section V.A, ψ_i in the left-hand sides of (66) $\sim$ (70) is interpreted as $\varphi_i[k+1]$. ψ_i and φ_i in the right-hand sides are interpreted as $\varphi_i[k]$ and $\varphi_i[k-1]$, respectively.

Figure 3 is Here.

Figure 3 illustrates the estimation of a data generating mechanism using the above Gaussian mixtures. Dots were generated by

$$y(t)=|x(t)|+n(t)$$

with $n(t)$ drawn from $\mathcal{N}(0.0,\,0.15^2)$ for $x(t)\in[-1.0,1.0]$. The set of pairs $\{(x(t),y(t))\}_{t=1}^{N}$ with $N=252$ is the incomplete data $(\mathcal{X},\mathcal{Y})$. The input $x(t)$ is applied to the Gaussian probability set (63) through the mean values

$$\mu_i^{\psi}(t)=[\theta_{i1}^{\psi},\theta_{i2}^{\psi}][1,x(t)]^T,\quad(i=1,2).$$

The dashed curve in Figure 3 is the estimated source mechanism

$$\overline{y^{\psi}(t)}=\sum_{i=1}^{2}g_i(x(t),\xi^{\psi})\mu_i^{\psi}(t),$$

where

$$g_i(x(t),\xi^{\psi})=\omega_i^{\psi}p(x(t)|\xi_i)/\{\textstyle\sum_{j=1}^{2}\omega_j^{\psi}p(x(t)|\xi_j)\}.$$

Figure 4 is Here.

Figure 4 illustrates convergence of the incomplete-data log-likelihood by using the α-EM algorithm. Table III summarizes this convergence trend. As is given in the fifth column, a CPU-time speedup ratio of 1.56 is obtained.

TABLE III is Here.

C. Practical Choice of α

Here, we consider guidelines for the practical choice of α. First, we list the theoretical results on the choice of α.

(i) Corollary 8 recommends the range of $\alpha>-1$ for the exponential family. The log-EM algorithm corresponds to $\alpha=-1$.

(ii) Corollary 9 also recommends $\alpha>-1$ if

$$-\partial^2\log p(y|\varphi^*)>0$$

is satisfied.

(iii) On the other hand, $\alpha<1$ is required because of the basic equality (26). This ensures positivity of the α-information matrix, due to (14).

Guided by the above theoretical results and more experiments in addition to those in Sections V.A and V.B, we propose the following practical strategies for choosing α.

(iv) $\alpha=0$ is a reasonable strategy for updates given in closed forms. The speed is fast enough and its computational load is lighter than general α. As was explained in Footnote 2, this case corresponds to the Hellinger distance.

(v) When update equations are in closed forms, we can start from $\alpha=1$ and then decrease α towards 0. This strategy was shown in Fig. 2 ($\alpha=1.0$ switched to 0.0). There can be many strategies on "when to switch" and "how to control."

(vi) In case that Newton-Raphson iterations are used, Hessian matrices of the following form appear [18], [19], [36].

$$H^{(\alpha)}(\psi|\varphi)=H_1(\psi|\varphi)+\left\{-\tfrac{1+\alpha}{2}H_2(\psi|\varphi)\right\}, \qquad (71)$$

where $H_i(\psi|\varphi)$, $(i=1,2)$, are positive definite matrices. Since the second term $-\frac{1+\alpha}{2}H_2(\psi|\varphi)$ is negative definite for $\alpha>-1$, the matrix $H^{(\alpha)}(\psi|\varphi)$ becomes a $(P+N^{(\alpha)})$ type for $\alpha\in(-1,1)$ which improves the convergence rate [7]. For stable speedup in this case,

we select any $\beta \in (-1, 1)$ and choose switching of $\alpha = \{-1, \beta\}$. Here, the case of $\alpha = -1$, i.e., the traditional log-EM family is used at iterations only if the positivity of $H^{(\beta)}(\psi|\varphi)$ is violated. Then, this method is convergent for $\alpha < 1$. We checked that this strategy works well and achieves speedup at least equivalent to Fig. 2 and Fig. 4. In this method, $\beta = 0$ is again a reasonable choice.

VI. Concluding Remarks

A new class of EM algorithms, named the α-EM family, was derived by the maximization transfer which uses more general surrogate functions than the logarithmic one. The log-EM family corresponds to the special case of $\alpha = -1$. The parameter α can however be selected to ameliorate the second-order properties of the surrogate functions. The α-EM algorithm converges faster than the log-EM algorithm in terms of both the number of iterations and the total computation time.

As we can observe from the literature on log-EM algorithms such as EM and GEM [1], ECM [27], ECME [28], SAGE [29], AECM [5], an important problem is the selection of augmented data. The log-EM family is quite rich, and new counterparts to such problems can be developed for the α-EM family. Also further exploration of practical issues pertaining to the α-EM family is needed.

Next, we would like to comment upon the possibility of other types of surrogate functions. Here, we focus our attention on the convex divergence (1) because of its general capacity on convex optimization [18]. As was explained in the text, the α-logarithm (6) started from the convex divergence. First, we summarize the properties of the α-logarithmic surrogate function.

(i) The parameter α controls the curvature of $L^{(\alpha)}(r)$ from "concave" to "convex." In its passing points, the logarithm $L^{(-1)}(r) = \log r$, which is concave, and the linear function $L^{(1)}(r) = r - 1$ exist.
(ii) The basic equality for the α-EM algorithm can be obtained exactly as (26). This equality specifies the use of the range $\alpha < 1$, i.e., only the concave α-log-likelihood ratio.
(iii) Prior distributions [4], [38] can be incorporated in the basic equality (26) [19].
(iv) For the range of $\alpha < 1$, the α-information matrix is positive and the Cramér-Rao bound is not degraded since Equation (14) holds.

For specific problems, some of the above properties or equivalents may be dropped. In such cases, there will be a similar but different class of surrogate functions which can show effective convergence. But, these functions still need to satisfy the concavity which corresponds to $\alpha < 1$, especially around $r = 1$.

Finally, we comment on the utilization of surrogate functions to other areas rather than the EM approach. A good example is independent component analysis, which uses the convex divergence as a direct surrogate function for independence. One such example is the f-ICA [39], [40]. The f-ICA, interpreted as a momentum and look-ahead update, yields remarkable speedup. This algorithm has been applied to brain functional-MRI mapping [40], [41].

[18] For instance, the integrand $qf(p/q)$ of (1) is regarded as a projective transformation of $f(p)$, which is related to conic quadratic programming [37].

[19] For instance, add or multiply $L^{(\gamma)}\left(w(\psi)/w(\varphi)\right), \gamma \in \boldsymbol{R}$, to both sides of (26). Here, $w > 0$ is a prior density or a penalty function.

Appendix A
Proof of Theorem 3

Since Ψ is a subset of $\boldsymbol{R}^d$, conditions (a), (c) and Lemma 1 show that $\{L_Y^{(\alpha)}(\varphi[n]|\varphi[0]),\ n \geq 0\}$ is bounded above for $\varphi[0] \in \Psi$. Condition (b) assures that the mapping $\mathcal{A}^{(\alpha)}$ is closed. Condition (d) assures that $\psi \in \Psi$ holds after differential operations on $Q_{X|Y,\varphi}^{(\alpha)}(\psi|\varphi)$ and $L_Y^{(\alpha)}(\psi|\varphi)$. Condition (e) is the key assumption which guarantees $L_Y^{(\alpha)}(\varphi[n+1]|\varphi[0])$ is monotone increasing due to Lemma 1 (vii). Then, the conditions for the global convergence theorem [25], [26] are met as follows:

(i) The sequence $\{\varphi[n]\}_{n\geq 0}$ is generated by $\varphi[n] \in \mathcal{A}^{(\alpha)}(\varphi[n-1])$.
(ii) $\mathcal{S}$ is contained in a compact set $\Psi_{\varphi[0]}$.
(iii) $L_Y^{(\alpha)}(\psi|\varphi)$ is continuous and

$$L_Y^{(\alpha)}(\varphi[n+1]|\varphi[0]) > L_Y^{(\alpha)}(\varphi[n]|\varphi[0]) \text{ for } \varphi[n] \in \Psi \backslash \mathcal{S},$$

where the left-hand side is bounded above.

Thus, Theorem 3 holds because of the global convergence theorem [25], [26]. □

Appendix B
Proof of Theorem 5

The proof uses techniques from [27], but we also need additional properties on the α-ECM algorithm. The roles of conditions (a) $\sim$ (d) are the same as in the proof of Theorem 3. Thus, the incomplete-data α-log-likelihood ratio for $\varphi[0]$ is monotone increasing and the α-ECM update is a closed mapping. Therefore, it suffices to show that a stationary point in $\mathcal{S}$ of the condition (g) is unconstrained. Let

$$\mathcal{J}(\varphi) \stackrel{\text{def}}{=} \{\psi \in \Psi \mid \partial^{10} L_Y^{(\alpha)}(\psi|\varphi) \in J(\varphi)\} \qquad (B.1)$$

where $J(\varphi)$ is defined by (39). Then, in the following way, we can show that $\varphi[n] \in \Psi \backslash \mathcal{J}(\varphi[n])$ implies

$$Q_{X|Y,\varphi}^{(\alpha)}(\varphi[n+1]|\varphi[n]) > 0.$$

Suppose not, i.e., suppose $\varphi[n] \in \Psi \backslash \mathcal{J}(\varphi[n])$ but

$$Q_{X|Y,\varphi}^{(\alpha)}(\varphi[n+1]|\varphi[n]) = 0.$$

Then, the uniqueness of the conditional maximizer gives

$$\varphi[n+1] = \varphi[n + (S-1)/S] = \cdots = \varphi[n + 1/S] = \varphi[n].$$

Therefore, $Q^{(\alpha)}_{X|Y,\varphi}(\psi|\varphi[n])$ is decreased along any feasible direction guided by $\Psi_s(\varphi[n])$ for all s. This means, from definition (37), that

$$\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n]|\varphi[n])\}^T \eta_s \leq 0$$

for all

$$\eta_s \in T_s(\varphi[n]);\ s = 1, \ldots, S.$$

Therefore, from definition (38), one obtains

$$\{\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n]|\varphi[n])\}^T \eta \leq 0$$

for all $\eta \in T(\varphi[n])$. Then, from definition (39), one obtains

$$\partial^{10} Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n]|\varphi[n]) \in J(\varphi[n]).$$

Thus, from (29) of Lemma 4, we have

$$\partial^{10} L^{(\alpha)}_{Y}(\varphi[n]|\varphi[n]) \in J(\varphi[n]).$$

This means, from (B.1), that $\varphi[n] \in \mathcal{J}(\varphi[n])$, which contradicts the assertion that $\varphi[n] \in \Psi \backslash \mathcal{J}(\varphi[n])$. Thus,

$$\varphi[n] \in \Psi \backslash \mathcal{J}(\varphi[n])$$

implies

$$Q^{(\alpha)}_{X|Y,\varphi}(\varphi[n+1]|\varphi[n]) > 0.$$

This assures

$$L^{(\alpha)}_{Y}(\varphi[n+1]|\varphi[n]) > 0.$$

Next, we use the space-filling condition $J(\varphi[n])) = \mathbf{0}$ of (40). By Lemma 1 (vii), the set $\mathcal{J}(\varphi[n])$ becomes the set of stationary points for $L^{(\alpha)}_{Y}(\varphi[n]|\varphi[0])$ with respect to

$$\varphi[n] \in \Psi \subset \boldsymbol{R}^d.$$

This is the set $\mathcal{S}$ itself in condition (g) of the α-ECM algorithm. □

ACKNOWLEDGMENT

The author is grateful to Prof. P. Moulin, the associate editor, for his constructive suggestions which improved this paper. The author wishes to thank the referees for their helpful comments. The inspiring papers on the log-EM family, e.g., EM, GEM, ECM, ECME, SAGE, AECM and others, are gratefully acknowledged. The author is also thankful to have worked in collaboration with his students, S. Furukawa (A & D), N. Takeda (SONY), T. Ikeda (NTT Data), and T. Niimoto (Toshiba).

REFERENCES

[1] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, pp. 1-38; with discussion by E.M.L. Beale, J.A. Nelder, C.A.B. Smith, R.J.A. Little, T.J. Orchard, B. Torsney, D.M. Titterington, G.D. Murray, D.A. Preece, K. Ord, M.J.R. Healy, L.E. Baum, W.H. Carter, B. Efron, S.E. Fienberg, I.Guttman, S.J. Haberman, H.O. Hartley, S.C. Pearce, S.R. Searle, R. Sundberg, E.A. Thompson, R. Thompson, B. Turnbull, and the authors' reply, pp.22-38, 1977.

[2] T. K. Moon, "The expectation-maximization algorithm," IEEE Signal Processing Society Magazine, pp. 47-60, November, 1996.

[3] M.A. Tanner, *Tools for Statistical Inference*, Third Edition, New York, Springer-Verlag, 1996.

[4] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, New York: Wiley-Interscience, 1997.

[5] X-L. Meng and D. van Dyk, "The EM algorithm – an old folk-song sung to a fast new tune," *J. R. Statist. Soc. B*, vol. 59, pp. 511-567; with discussion by D.B. Rubin, D.M. Titterington, W.R. Gilks, J. Diebolt, M. Aitkin, C.A.B. Smith, J. Hinde, J.T. Kent, D.E. Tyler, P. Damien, D. Chauveau, D. Draper, J.A. Dupuis, J. Fessler, A. Gelman, P.J. Green, A.O. Hero, III, M. Lavielle, C. Liu, J.S. Liu, G.O. Roberts, S.K. Sahu, A.M. Zaslavsky, and the authors' reply, pp. 541-567, 1997.

[6] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Section 8.3 (d), New York, Academic Press, 1970.

[7] K. Lange, D.R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. of Computational and Graphical Statistics*, vol. 9, pp. 1-59; with discussion by H.A.L. Kiers, J. de Leeuw, G. Michailidis, Y.N. Wu, X.-L. Meng, P.J.F. Groenen, W.J. Heiser, and A. Gelman, and with rejoinder by D.R. Hunter and K. Lange, pp. 21-59, 2000.

[8] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, New York: Wiley-Intersciences, 1991.

[9] A. Rényi, "On measures of entropy and information," *Proc. 4th Berkeley Symp. Math. Stat. and Pr.*, vol. 1, pp. 547-561, 1960.

[10] J. Havrda and F. Charvát, "Qualification method of classification processes: Concept of structural α-entropy," Kybernetika, vol. 3, pp. 30-35, 1967.

[11] I. Csiszár, "Information-type measures of difference of probability distributions and indirect observations," *Studia Sci. Math. Hungarica*, vol. 2, pp. 299-318, 1967.

[12] S. Amari, "Differential geometry theory of statistics," in S.-I. Amari, O.E. Barndorff-Nielsen, R.E. Kass, S.L. Luritzen, and C.R. Rao (Eds.), *Differential Geometry in Statistical Inference*, Institute of Mathematical Statistics Lecture Notes, vol. 10, pp. 21-94, 1985.

[13] J.N. Kapur and H.K. Kesavan, *Entropy Optimization Principles with Applications*, San Diego: Academic Press, 1992.

[14] S. Amari and H. Nagaoka, *Methods of Information Geometry*, Tokyo: Iwanami, 1993 (Translation by D. Harada, Providence: AMS, 2000).

[15] A.N. Iusem, B.F. Svaiter, and M. Teboulle, "Entropy-like proximal methods in convex programming," Mathematics of Operations Research, vol. 19, pp. 790-814, 1994.

[16] Y. Matsuyama, "The weighted EM learning and monitoring structure," in *Information Processing Society of Japan, 54th Convention Record*, 6G-04, March, 1997.

[17] Y. Matsuyama, "The α-EM algorithm: A block connectable generalized learning tool for neural networks," Lecture Notes in Computer Science, No. 1240, pp. 483-492, Berlin, Germany: Springer-Verlag, June, 1997.

[18] Y. Matsuyama, "The weighted EM algorithm and block monitoring," in *Proc. International Conference on Neural Networks*, vol. 3, pp. 1936-1941, June, 1997.

[19] Y. Matsuyama, "Non-logarithmic information measures, α-EM algorithms, and speedup of learning," in *Proc. International Symposium on Information Theory*, p. 385, Cambridge, Mass., August 1998.

[20] Y. Matsuyama, "The α-EM algorithm and its basic properties," *Trans. Inst. Electro. Info. Comm. Engr.*, vol. J82-D-I, pp. 1347-1358, 1999.

[21] I.A. Ibragimov and R.Z. Has'minskii, *Statistical Estimation: Asymptotic Theory*, New York: Springer-Verlag, 1981.

[22] M.J. Schervish, *Theory of Statistics*, New York: Springer-Verlag, 1995.

[23] G.B. Folland, *Real Analysis: Modern Techniques and Their Applications*, Second Edition, New York, John Wiley & Sons, 1999.

[24] C.F.J Wu, "On the convergence properties of the EM algorithm," *Annals of Statistics*, vol. 11, pp. 95-103, 1883.

[25] W.I. Zangwill, *Nonlinear Programming: A Unified Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1969.

[26] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley, 1973.

[27] X.L. Meng and D.B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, pp. 267-278, 1993.

[28] C. Liu and D.B. Rubin, "The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence," *Biometrika*, vol. 81, pp. 633-648, 1994.

[29] J.A. Fessler and A.O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Processing*, vol. 42, pp. 2664-2677, 1994.

[30] R.A. Redner and H.F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, pp. 195-239, 1984.

[31] L. Xu and M.I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation*, vol. 8, pp. 129-151, 1995.

[32] Y. Matsuyama, "Harmonic competition: A self-organizing multiple criteria optimization," *IEEE Trans. Neural Networks*, vol. 7, pp. 652-668, 1996.

[33] Y. Matsuyama, "Multiple descent cost competition: Restorable self-organization and multimedia information processing," *IEEE Trans. Neural Networks*, vol. 9, pp. 106-122, 1998.

[34] Y. Matsuyama, N. Takeda, S. Furukawa and T. Niimoto, "A hierarchy from α-EM algorithm to vector quantization and self-organization," in *Proc. International Conference on Neural Information Processing*, vol. 1, pp. 233-238, Kita-Kyushu, Japan, October 1998.

[35] L. Xu, M.I. Jordan and G.E. Hinton, An alternative model for mixtures of experts, in *Advances in Neural Information Processing*, The MIT Press, vol. 7, pp. 633-640, 1995.

[36] Y. Matsuyama, S. Furukawa, N. Takeda and T. Ikeda, "Fast α-weighted EM learning for neural networks of module mixtures," in *Proc. International Joint Conference on Neural Networks*, vol. 3, pp. 2306-2311, Anchorage, Alaska, May 1998.

[37] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, p. 96, Philadelphia: SIAM and MPS, 2001.

[38] J.A. Fessler and A.O. Hero, III, "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithm," *IEEE Trans. Image Processing*, vol. 4, pp. 1417-1438, 1995.

[39] Y. Matsuyama, N. Katsumata, Y. Suzuki and S. Imahara, "The α-ICA Algorithm," in *Proc. Second International Workshop on Independent Component Analysis and Blind Signal Separation*, pp. 297-302, Helsinki, Finland, June 2000.

[40] Y. Matsuyama, S. Imahara and N. Katsumata, "Optimization transfer for computational learning: A hierarchy from f-ICA and alpha-EM to their offsprings," *Proc. International Joint Conf. Neural Networks*, vol. 3, pp. 1883-1888, Hawaii, May 2002.

[41] Y. Matsuyama and R. Kawamura, "Supervised map ICA: Applications to brain functional MRI," in *Proc. International Conf. Neural Information Processing*, vol. x, pp. y-z, Singapore, November 2002.

Yasuo Matsuyama (S'77-M'78-SM'92-F'98) received the B. Eng., M. Eng. and Dr. Eng. degrees in 1969, 1971, 1974, respectively, all from Waseda University, Japan. In 1978, he received the Ph.D. degree from Stanford University, CA, under the Personnel Exchange Program of the Japan Society of the Promotion of Science.

Since 1996, he has been a Professor at Waseda University. Formerly, he was a Professor and Division Chairperson of the doctor course at Ibaraki University. In 1994, he was with the National Personnel Authority as a Cochairperson of the governmental personnel selection.

His current research interests include computational and communication mechanisms, symbol/subsymbol learning algorithms, multimedia information processing, biological information processing, and their electronic implementations.

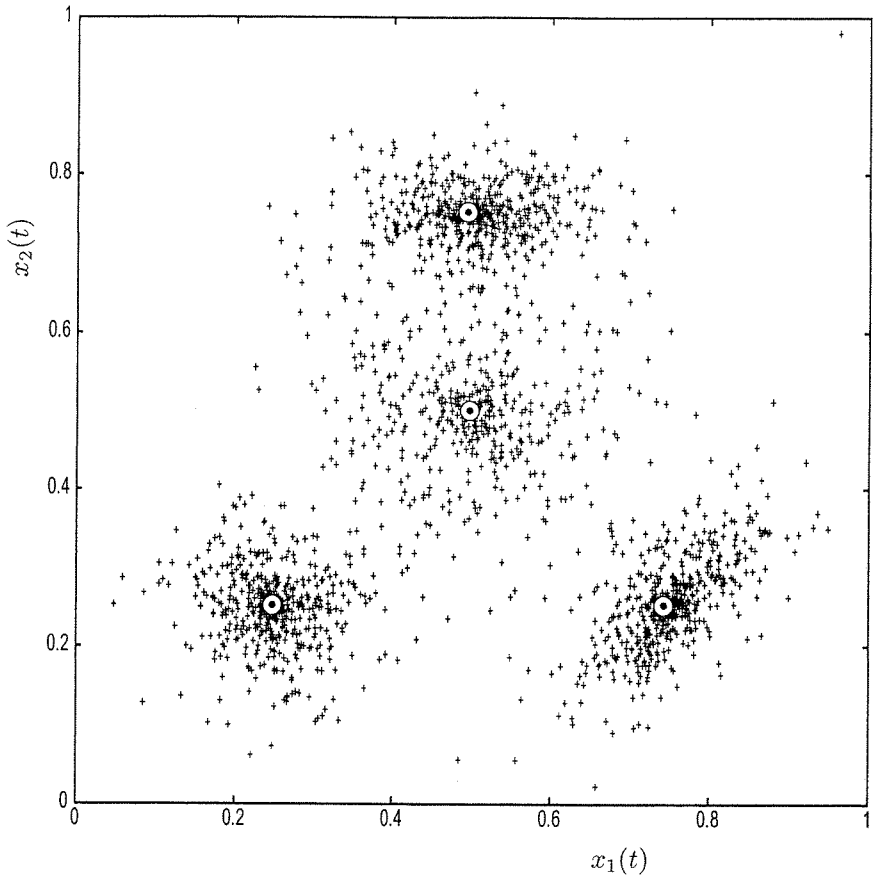
Dr. Matsuyama received the Outstanding Paper Award from the IEEE Neural Networks Society, and the best paper awards form the Institute of Electronics, Information and Communication Engineers (IEICE), and from the Telecommunications Foundation. He is a Fellow of the IEICE, where he was a Councilor of the Tokyo Chapter from 1995 to 1996.

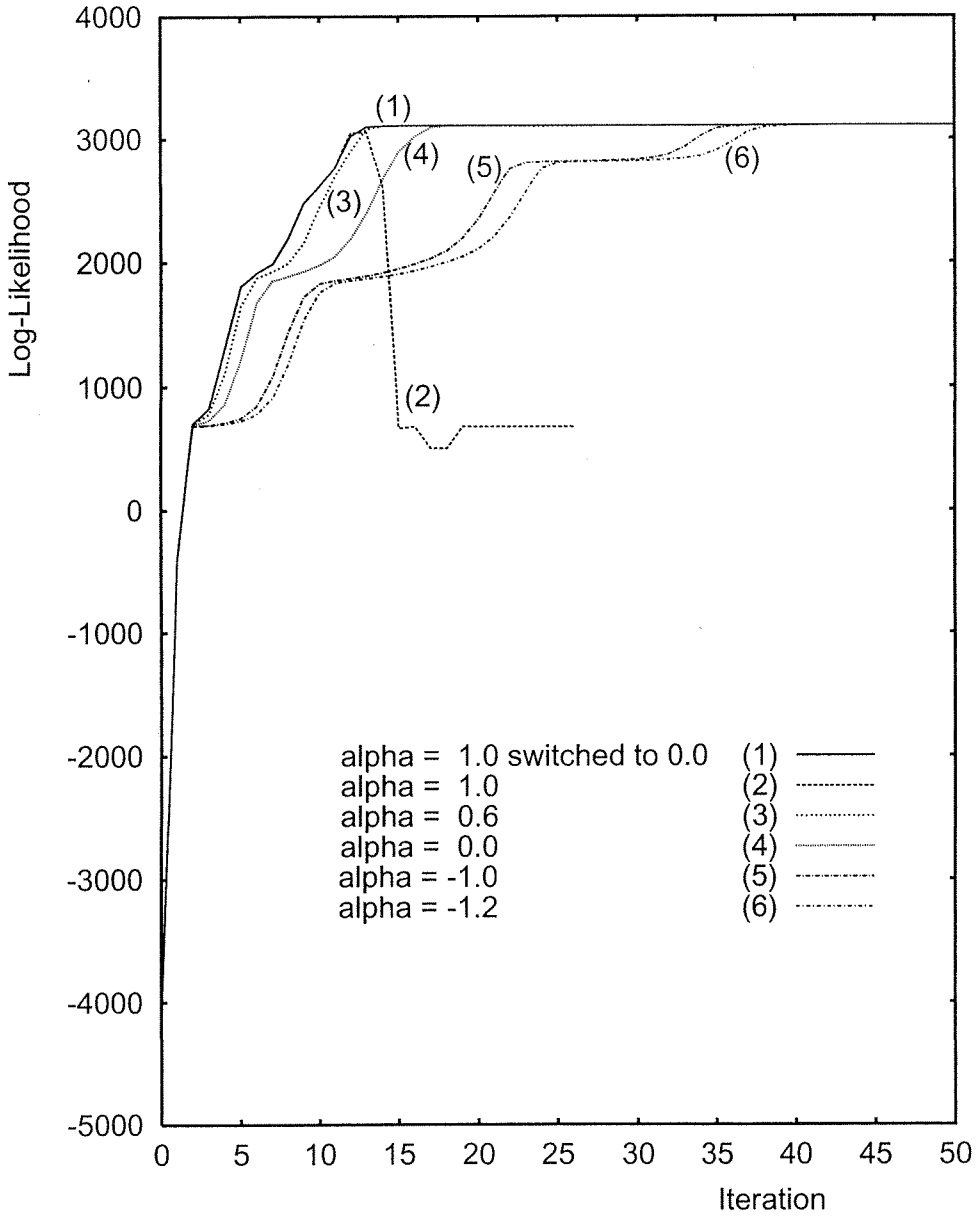
Figure Captions

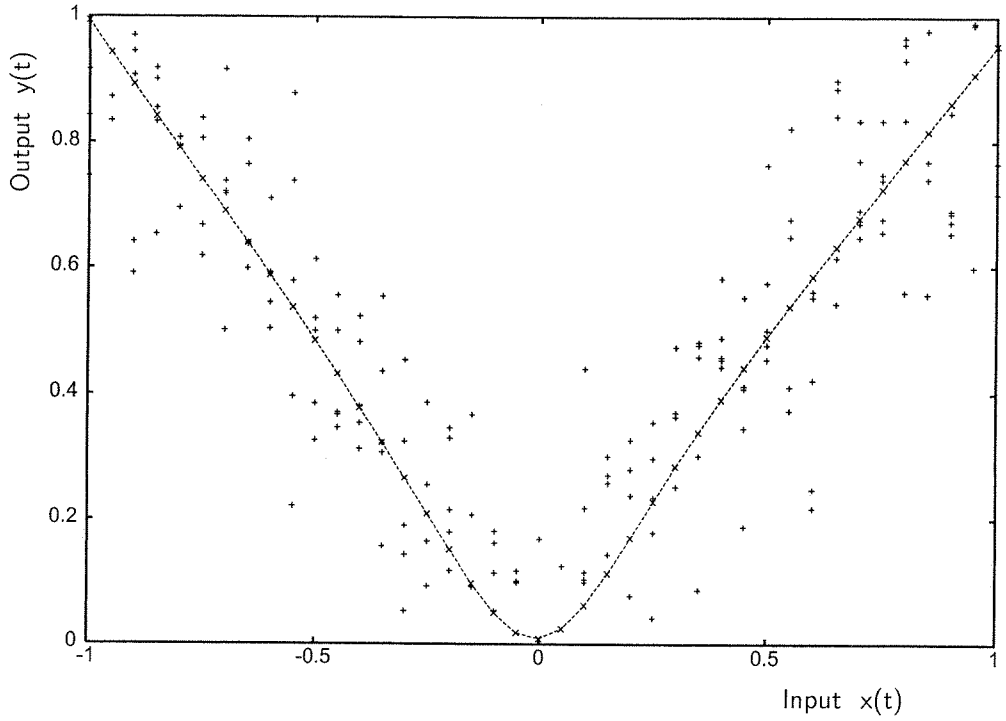
The α-EM Algorithm: Surrogate Likelihood Maximization Using α-Logarithmic Information Measures

by Yasuo MATSUYAMA (CLN: 98-476)

Fig. 1. Gaussian mixtures and obtained mean vectors.
Fig. 2. Convergence of clustering for various α.
Fig. 3. Observed data and source estimation.
Fig. 4. Convergence of source estimation for various α.


1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
$x_2(t)$
$x_1(t)$


4000
3000
2000
1000
0
-1000
-2000
-3000
-4000
-5000
Log-Likelihood
0
5
10
15
20
25
30
35
40
45
50
Iteration
(1)
(2)
(3)
(4)
(5)
(6)
alpha = 1.0 switched to 0.0 (1)
alpha = 1.0 (2)
alpha = 0.6 (3)
alpha = 0.0 (4)
alpha = -1.0 (5)
alpha = -1.2 (6)


Output y(t)
Input x(t)
1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1

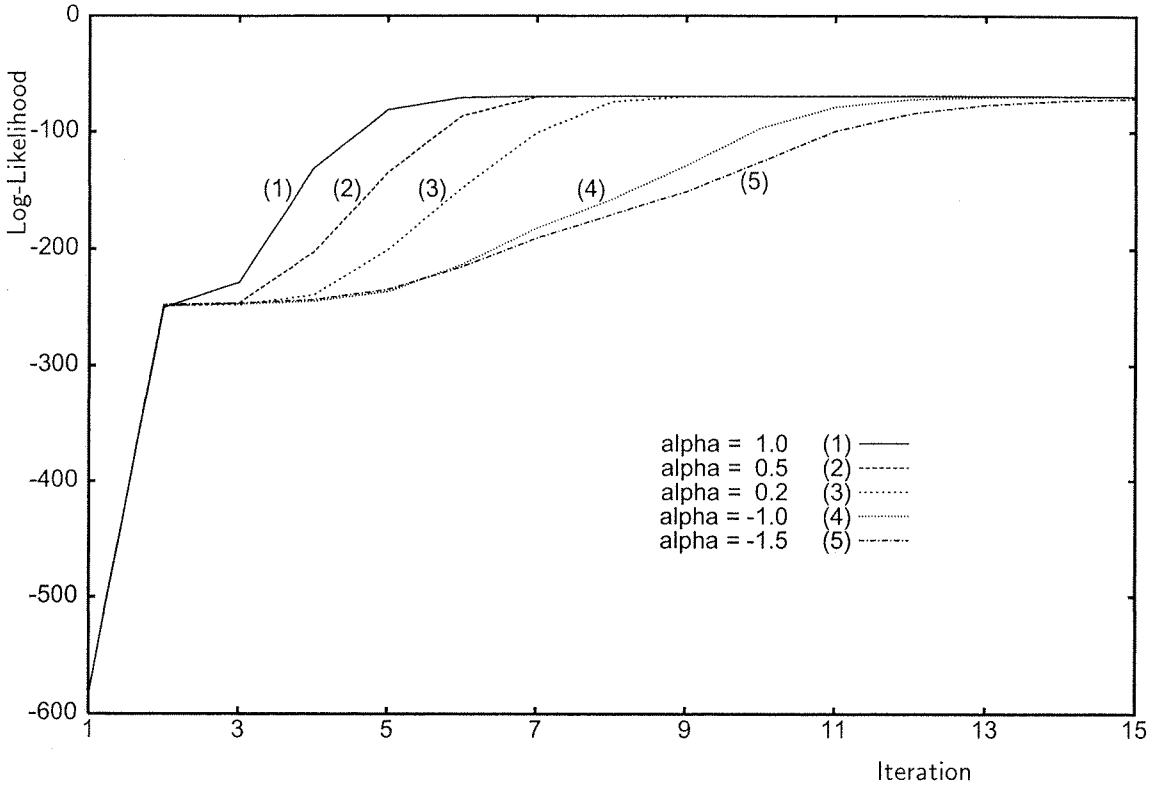

0
-100
-200
-300
-400
-500
-600
Log-Likelihood
1
3
5
7
9
11
13
15
Iteration
(1)
(2)
(3)
(4)
(5)
alpha = 1.0 (1)
alpha = 0.5 (2)
alpha = 0.2 (3)
alpha = -1.0 (4)
alpha = -1.5 (5)

Table Titles

The α-EM Algorithm: Surrogate Likelihood Maximization Using α-Logarithmic Information Measures

by Yasuo MATSUYAMA (CLN: 98-476)

Data	α-EM Family	Gaussian Mixture Clustering
incomplete data; $\mathcal{I}$	y	$\{x(t)\} \stackrel{\text{def}}{=} \mathcal{X}$
missing data; $\mathcal{Z}$	implicit	$\{z(t)\} = \mathcal{Z}$
complete data; $\mathcal{C}$	x	$(\{x(t)\}, \{z(t)\}) = (\mathcal{X}, \mathcal{Z})$

α	Iterations	Iteration Speedup Ratio	CPU-Time Ratio	CPU-Time Speedup Ratio
-1.00	37	1.00	1.00	1.00
$+0.60$	14	2.64	1.44	1.84
$+1.00 \rightarrow 0.00$	14	2.64	1.44	1.84

α	Iterations	Iteration Speedup Ratio	CPU-Time Ratio	CPU-Time Speedup Ratio
-1.00	15	1.00	1.00	1.00
$+0.50$	9	1.67	1.07	1.56